

Computational Intelligence Algorithms for Optimisation of Wireless Sensor Networks



A thesis submitted in partial fulfilment of the requirements for the
Degree of Doctor of Philosophy

BY

Muyiwa Olakanmi Oladimeji

Student ID 3023261

Supervised by

Dr Sandra Dudley

School of Engineering, London South Bank University

March, 2017

I would like to dedicate this thesis to my loving parents and siblings for always being there to guide me throughout my childhood and adult years, for encouraging, praising and loving me. I grew strong with your support, dependable and genuine love. Thank you all for your priceless affection. And to my fiancée Taiwo, thank you for making me smile and laugh in times of frustration.

Acknowledgements

Foremost, I thank Almighty God for granting me the grace to complete this thesis. I would like to offer my sincere thanks to my parents for their continuous support. Also, very important I would like to express my sincere gratitude to my supervisor Dr. Sandra Dudley for her timely suggestions, coordination and persistent supervision. I would additionally like to thank Prof. Ghavami and Dr. Mikdam Turkey for their support. Finally, I must thank all of my colleagues for their encouragement and moral support.

Abstract

Recent studies have tended towards incorporating Computation Intelligence, which is a large umbrella for all Machine Learning and Meta-heuristic approaches into wireless sensor network (WSN) applications for enhanced and intuitive performance. Meta-heuristic optimisation techniques are used for solving several WSN issues such as energy minimisation, coverage, routing, scheduling and so on. This research designs and develops highly intelligent WSNs that can provide the core requirement of energy efficiency and reliability. To meet these requirements, two major decisions were carried out at the sink node or base station. The first decision involves the use of supervised and unsupervised machine learning algorithms to achieve an accurate decision at the sink node. This thesis presents a new hybrid approach for event (fire) detection system using k -means clustering on aggregated fire data to form two class labels (fire and non-fire). The resulting data outputs are trained and tested by the Feed Forward Neural Network, Naive Bayes, and Decision Trees classifier. This hybrid approach was found to significantly improve fire detection performance against the use of only the classifiers. The second decision employs a metaheuristic approach to optimise the solution of WSNs clustering problem. Two metaheuristic-based protocols namely the Dynamic Local Search Algorithm for Clustering Hierarchy (DLSACH) and Heuristics Algorithm for Clustering Hierarchy (HACH) are proposed to achieve an evenly balanced energy and minimise the net residual energy of each sensor nodes. This thesis proved that the two protocols outperforms state-of-the-art protocols such as LEACH, TCAC and SEECH in terms of network lifetime and maintains a favourable performance even under different energy heterogeneity settings.

Contents

Contents	iv
List of Figures	viii
List of Tables	x
Nomenclature	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Research Objective	3
1.3 Thesis Contributions	4
1.4 Journal and Conference Publications	5
1.5 Thesis Organization	5
2 Energy Efficiency Mechanisms for WSNs	7
2.1 Introduction	7
2.1.1 Applications of WSNs	9
2.1.2 Requirements and Challenges of WSNs Design	11
2.1.3 Energy Consumption in WSNs	14
2.2 Sleep Scheduling Mechanisms Overview	15
2.3 Clustering Mechanisms Overview	18
2.3.1 Low Energy Adaptive Clustering Hierarchy (LEACH)	18
2.3.2 Topology Controlled Adaptive Clustering (TCAC)	20
2.3.3 Scalable Energy Efficient Clustering Hierarchy (SEECH)	22

2.3.4	Other Clustering Approaches	25
2.4	Metaheuristic Algorithms	26
2.4.1	Global Search Strategy	27
2.4.2	Local Search Strategy	33
2.4.3	Memetic Algorithm	37
2.5	Clustering using Meta-Heuristic Algorithms	38
2.5.1	Clustering Using Genetic Algorithm	39
2.5.2	Clustering using Particle Swarm Optimisation	41
2.5.3	Clustering using Differential Evolution	41
2.6	Conclusion	42
3	Intelligent Machine Learning Mechanisms for WSNs	43
3.1	Introduction	43
3.2	Intelligent WSN-based Approach for Event Application	47
3.3	Supervised learning Algorithms	48
3.3.1	Artificial Neural Network	48
3.3.2	Naive Bayes Classifier	49
3.3.3	Decision Tree	50
3.4	Unsupervised Learning (Clustering) Algorithm	51
3.4.1	Fixed-width clustering	51
3.4.2	k -means Clustering	52
3.5	Conclusion	53
4	Proposed Machine Learning Approaches for WSNs	54
4.1	Introduction	54
4.2	Background	56
4.3	Data Aggregation in Clustered-Based WSNs	57
4.4	Proposed Hybrid Learning Approach	59
4.5	Empirical Results	65
4.6	Conclusion	66
5	Proposed Dynamic Local Search-Based Algorithm	67
5.1	Introduction	68
5.2	Network and Radio Model	71

CONTENTS

5.3	Proposed Objective function	72
5.4	The Proposed DLSACH protocol	73
5.4.1	Proposed SSIN Mechanisms	75
5.4.2	Proposed ILSACHS protocol	77
5.5	Energy Consumption Computation	80
5.5.1	Set-up Phase	81
5.5.2	Steady State Phase	82
5.6	Performance Evaluation	84
5.6.1	Performance Measures	86
5.6.2	Results and Discussion	86
5.7	Conclusion	89
6	Proposed Global-based Search Algorithm	91
6.1	Introduction	91
6.2	The proposed HACH Protocol	92
6.2.1	Clustering Operations using HEECHS protocol	94
6.2.2	Proposed Heuristic Crossover	96
6.2.3	Other Operators	98
6.3	Performance Evaluation	99
6.3.1	Stability Period and Network Lifetime	101
6.3.2	Average Energy at First Node Dies (AEFND)	103
6.3.3	WSNs Heterogeneity	104
6.3.3.1	Full heterogeneity	106
6.3.3.2	Partial heterogeneity	106
6.4	Conclusion	108
7	Conclusion and Recommended Future Work	109
7.1	Thesis Conclusion	109
7.2	Recommended Future Work	110
	References	112
	MATLAB Source Codes	125
A	Genetic Algorithm Operators	125

CONTENTS

A.1	Initialization	125
A.2	Tournament Selection	126
A.3	Heuristic Crossover	127
A.4	Mutation	130
B	Proposed SSIN protocol	130
C	Members	134
D	Objective Function	135
E	Proposed HACH protocol	137
F	Proposed DLSACH protocol	141
G	Energy Consumption	143
H	Main Code	144

List of Figures

2.1	The components of Sensor node	9
2.2	Flowchart for genetic algorithm Kachitvichyanukul [2012]	28
2.3	Flowchart for Particle Swarm Optimisation algorithm Kachitvichyanukul [2012]	29
2.4	Flowchart for differential evolution Kachitvichyanukul [2012]	32
4.1	Cluster based sensor network	58
4.2	Diagram of Feed Forward Neural Network	59
4.3	Density Curves of CO, Temperature, Photoelectric and Ionization respectively.	63
4.4	Diagram of Decision Tree	64
5.1	Radio Energy Dissipation Model	72
5.2	Covered Grid points in a 10×8 Sensing field	75
5.3	Illustration of Nodes to Sleep on Coverage Area	76
5.4	Step Size division of Search Length	79
5.5	The operational sequence of the proposed clustering protocols	83
5.6	Network Lifetime Comparison of DLSACH with LEACH, SEECH, TCAC	88
6.1	One round of the clustering process	94
6.2	Sensor nodes Topology and Random distribution	95
6.3	Binary representation of individuals in the population	97
6.4	Lifetime evaluation of HACH, LEACH, SEECH and TCAC	102

LIST OF FIGURES

6.5	Average residual energy of nodes alive versus rounds (refer to Exp_{ROM100})	104
6.6	Round number versus numbers of heterogeneous sensors	105
6.7	Performance Comparison of different WSNs Heterogeneity Level for (a.) FND, (b.) LND and (c.) IPL measures.	107

List of Tables

4.1	Distribution table for the four sensor types	63
4.2	Empirical Results for all Classifiers	65
5.1	Communication Parameters with Specified Values	85
5.2	Parameter values for each experiment	85
5.3	Comparison of LEACH, TCAC, SEECH and DLSACH for FND,LND and IPL	87
5.4	Performance Measures for Experiment III,IV and V	87
6.1	Parameter settings for Homogeneous WSNs Scenarios	99
6.2	Parameter settings for Heterogeneous WSNs Scenarios	100
6.3	Performance comparison of LEACH, TCAC and SEECH with HACH	101
6.4	AEFND of proposed HACH protocol	103
6.5	Performance Measures for different heterogeneous WSN Scenarios	105

Nomenclature

Acronyms

DLSACH	Dynamic Local Search Algorithm for Clustering Hierarchy
HACH	Heuristics Algorithm for Clustering Hierarchy
HEECHS	Heuristic Enhance Evolutionary Algorithm for Cluster Head Selection
SSIN	Stochastic Selection of Inactive Node
CHs	Cluster-heads
CI	Computaional Intelligence
DE	Differential Evolution
DT	Decision Tree
FFNN	Feed-Forward Neural Network
GA	Genetic Algorithms
GD	Gradient Descent
HEED	Hybrid Energy Efficient Distributive
ILS	Iterated Local Search
LDS	Linear Distance-based Scheduling
LEACH	Low Energy Adaptive Clustering Hierarchy

LEACH-C	LEACH-Centralised
MA	Memetic Algorithm
ML	Maximum Likelihood
NBC	Naive Bayes Classifier
NFDRS	National Fire Danger Rating System
NN	Neural Network
PSO	Particle Swarm Optimisation
RS	Randomized Scheduling
SA	Simulated Annealing
SEECH	Scalable Energy Efficient Clustering Hierarchy
TCAC	Topology-Controlled Adaptive Clustering
TS	Tabu Search
WSNs	Wireless Sensor Networks

Symbols

Acc_{eff}	Accumulated coverage effect
$AvgECH$	Average energy of CHs
$AvgENCH$	Average energy of non-CHs
C_{eff}	Coverage effect
d_0	Threshold distance
E_{Aggreg}	Energy consumed in aggregating measured data
E_{CH}	Energy consumed by Cluster heads
E_{elect}	Electronics Energy

E_{Mem}	Energy consumed by Member nodes
E_{Res}	Residual Energy
E_{Rx}	Energy consumed by the transceiver to receive a data message
E_{Total}	Total energy consumed by a wireless node
E_{Tx}	Energy consumed by the transceiver to send a data message
k	Packets
k_{CP}	Control packets
L	Number of Cluster heads
m_p	mutation probability
Max_{eff}	Maximum coverage effect
R	Risk penalty
R_s	Sensing range
w_1, w_2	Weighting factors

Chapter 1

Introduction

1.1 Motivation

Recent progress in Micro-electromechanical Systems (MEMS) has enabled the development of self-configurable and spatially distributed autonomous sensors [Naeimi et al. \[2012\]](#). These sensor nodes can be networked and deployed randomly in remote and inaccessible areas, hence producing useful wireless sensor networks (WSNs). In large areas, WSNs are used for gathering data from the sensor field and transmitting data to a distant sink. The potential applications of WSNs are environmental monitoring, target field imaging, weather monitoring, security, battlefield surveillance, event detection etc. The event detection is a newly discovered WSN functionality that offers extended capability of reporting data that contain time and location of events, which is contrast to periodic monitoring that transfers data without any abnormal change of condition. The design of in-network event detection methods for wireless sensor networks is not an easy task, as there is a need to cope with various challenges and issues such as the unreliability, heterogeneity, adaptability and most especially resource constraints such as battery energy.

Energy efficiency and network lifetime are major issues that require consideration in the design of protocols for WSNs. The field of WSNs embrace innovative techniques that can eliminate energy inefficiencies that would shorten the network lifetime. Energy constraint is a major problem in WSNs most especially when

larger number of sensor nodes are deployed. The limited energy poses many challenges to the design and management of WSNs and necessitates energy-awareness at all layers of the networking protocol stack. For example, at the network layer, it is highly desirable to find methods for energy-efficient route discovery and relaying of data from the sensor nodes to the base station (BS) so that the lifetime of the network is maximized [Abbasi and Younis \[2007\]](#). Most existing routing protocols designed to tackle the above challenges are broadly classified into two classes, namely flat and hierarchical. Flat protocols include the old-fashion Direct Transmission (DT) and Minimum Transmission Energy (MTE), which cannot promise a balanced distribution of the energy among sensors in a WSN. The drawback with DT is that sensor nodes communicate directly with the sink and this causes far away sensors to die first. In the MTE, far away sensors use a relay sensor for data transmission to the BS and this causes the relay sensor to die first.

Therefore, designing energy-efficient clustering protocols becomes a major factor for lifetime extension of sensors. Generally, clustering protocols can outperform flat protocols in balancing energy consumption and network lifetime prolongation by adopting data aggregation mechanisms [Abbasi and Younis \[2007\]](#); [Heinzelman et al. \[2002\]](#). Theoretically, there are three types of nodes, namely the cluster-head (CH), member node (MN) and sink node (SN). The member node is responsible for sensing the raw data and employs TDMA scheduling to send the raw data to the CH. The main role of the CH is to aggregate data received from member nodes (MN) and thereby forwards the aggregated data to the sink through single-hop or multi-hop. CH selection can either be done by the sensors themselves, by the BS or can be pre-determined by the wireless network designer. From a theoretical and practical point of view, WSNs can be classified into the Homogeneous (WSNs with the same sensor node configuration e.g. Energy) and the Heterogeneous (WSNs with dissimilar sensor node configuration).

Reliability is another key issue that needs to be considered for some critical or event detection applications such as indoor fire detection, forest fire and pipeline monitoring etc. Fast and accurate fire detection helps to minimise fire losses that often results into loss of lives and damage to properties. Therefore, researchers have been investigating new techniques that will help in fast and accurate fire detection. For a system to decide accurately an abnormal condition such as fire,

there maybe a requirement to combine several attributes based on large number of sensor types (temperature, carbon monoxide (CO), smoke) which are spatially distributed over a wide area Memon and Muntean [2012]. Data obtained from a composite event are multidimensional in nature. One of the key measures of enhancing accurate fire detection decisions is to perform data aggregation at intermediate nodes or at the cluster head. Data aggregation usually involves the fusion of data from multiple sensors at intermediate nodes and transmission of the aggregated data to the base station (sink). Data aggregation helps to remove redundant and highly correlated data generated from neighbouring sensors at the intermediate node before transmission to the base station Memon and Muntean [2012]. Data aggregation techniques are also very effective in reducing communication overhead by collecting the most critical data from the sensors and making it available to the sink in an energy efficient manner with minimum data latency. Data latency is a crucial requirement in most event detection application such as fire detection applications.

1.2 Research Objective

The main objective of this thesis is to analyse, investigate applicability and optimise computational intelligence methods for energy-efficient and intelligent operations of wireless sensor networks (WSNs). Therefore, in order to achieve the objectives, the research sub-objectives of this thesis are as follows:

1. Design a new protocol that is based on metaheuristic algorithms that can equally and efficiently distribute the energy consumptions evenly among sensor nodes and still achieve an extended network lifetime compared with the state-of-the-art designs often offered as solutions for a clustered architecture in the literature.
2. Provide a comparative assessment of other state-of-the-art protocols with the new design protocol for lifetime extension purpose using a simulated environment.
3. Present a new hybrid approach based on machine learning algorithm that

can efficiently extract patterns and detect trends that are hidden in complex fire data sets. This objective aims at improving the detection accuracy of fire detection systems compared with the current state-of-the-art approaches that has been used for the similar problem.

4. Provide a comparative study of state-of-the-art event detection techniques in terms of their detection rate and accuracy.

1.3 Thesis Contributions

The contributions of this thesis are:

1. The design and implementation of an energy-efficient transmission protocol for extending lifetime of wireless sensor networks.
2. The development of a sleep scheduling mechanism that is based on the Boltzmann selection process in genetic algorithms for conserving the energy of sensor nodes. The network coverage is analysed and put into considerations in the selection of inactive nodes.
3. The design of new WSN clustering protocol that employs metaheuristic approaches to distribute cluster head and energy loads evenly among sensor for the purpose of prolonging WSNs lifetime.
4. Perform data aggregation on three multi-dimensional datasets obtained from a real time fire scenario.
5. The proposal of new hybrid machine learning approaches for accurate event detection using k -means and other classification models.

1.4 Journal and Conference Publications

The following publications have resulted from various chapters of this thesis

1. **Muyiwa O. Oladimeji**, Mikdam Turkey, Mohammed Ghavami and Sandra Dudley, "A New Approach for Event Detection using k-means Clustering and Neural Networks", IEEE International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015, Pages 1-5. DOI: 10.1109/IJCNN.2015.7280752
2. **Muyiwa O. Oladimeji**, Mikdam Turkey and Sandra Dudley, "A Heuristic Crossover Enhanced Evolutionary Algorithm for Clustering Wireless Sensor Network", EvoApplications 2016, Porto, Portugal, March 30 –April 1, 2016, Proceedings, Part I, Pages 1-16, 2016a. DOI: 10.1007/978-3-319-31204-017
3. **Muyiwa O. Oladimeji**, Mikdam Turkey and Sandra Dudley, "Iterated Local Search Algorithm for Clustering Wireless Sensor Networks", IEEE Congress on Evolutionary Computation, CEC 2016, Vancouver, BC, Canada, July 24-29, 2016, Pages 3246-3253, 2016b. DOI: 10.1109/CEC.2016.7744200
4. **Muyiwa O. Oladimeji**, M. Turkey and S. Dudley, "HACH: Heuristic Algorithm for Clustering Hierarchy Protocol in Wireless Sensor Networks", Applied Soft Computing Journal, Volume 55, Pages 452–461, 2017. DOI: 10.1016/j.asoc.2017.02.016

1.5 Thesis Organization

The rest of this thesis is organised as follows:

Chapter 2 reviews wireless sensor networks (WSNs) and these applications, requirements, challenges and energy consumption. The chapter reviews the recent sleep scheduling and clustering mechanisms for the design of energy-efficient WSNs. As a subset of computational intelligence, a review on the different metaheuristic optimisation algorithms as presented in this chapter. Finally, the chapter reviews optimisation of energy-efficient cluster-based WSNs using metaheuristic approaches.

Chapter 3 introduces the use of WSNs for monitoring or event detecting applications such as fire detection system. It discusses the problem of traditional fire detection techniques and the introduction of WSNs into fire detection systems. It discusses new trend of incorporating artificial intelligence-based techniques into WSNs-based fire detection system for improved performance. Finally the chapter reviews various AI-based techniques under the machine learning approaches for fire detection applications.

Chapter 4 proposes a new hybrid approach to event detection that combines data aggregation, k -means clustering and supervised machine learning approaches such as feed forward neural network (FFNN), Naïve Bayes (NB), Decision Tree (DT).

Chapter 5 presents a new local-based metaheuristic approach for energy optimisation in WSNs called Dynamic Local Search Algorithm for Clustering Hierarchy (DLSACH). Under the DLSACH algorithm, the Stochastic Selection of Inactive Nodes (SSIN) and Iterated Local Search Algorithm for Cluster Head Selection (ILSACHS) protocols are proposed and they work cooperatively to minimise the energy consumption and extend the WSNs lifetime. The algorithms are evaluated via simulation experiments and compared with some existing algorithms.

Chapter 6 propose the Heuristics Algorithm for Clustering Hierarchy (HACH). It introduce the SSIN mechanism proposed in the previous chapter for sleep scheduling operation and a novel heuristic crossover operator to combine two different solutions to achieve an improved solution that enhances the distribution of cluster head nodes and coordinates energy consumption in WSNs. It presents the performance in terms of lifetime extension under various WSNs conditions.

Chapter 7 closes the thesis, reviewing the work undertaken and draws conclusions about key parts of the work presented. Finally, future work is discussed.

Chapter 2

Energy Efficiency Mechanisms for WSNs

This chapter presents a background on wireless sensor networks and the state-of-the-art energy saving mechanisms in WSNs. It covers aspects of WSNs such as components, applications, prominent challenges and energy consumption models. A review on the design energy-efficient WSNs using all the different classifications of sleeping and clustering techniques is discussed. Also, a review on the different meta-heuristic optimisation algorithm, which are the subset of computational intelligence. Lastly, this chapter presents a review on the design of cluster-based energy efficient WSNs using meta-heuristic search strategies.

2.1 Introduction

Wireless Sensor Networks (WSNs) have grown to be a powerful technological platform with vast and profound applications. They have transformed into an important technology with many simple and complex applications such as environment monitoring, surveillance systems and military operations. A WSN usually consists of tens to thousands of sensor nodes that communicate via wireless medium for the purpose of sharing and processing data [Yu et al. \[2006\]](#). Sensor nodes are usually deployed randomly in a sensor field. They wirelessly communicate with each other to coordinate themselves in order to produce reliable and

2. Energy Efficiency Mechanisms for WSNs

precise information about the physical environment under their coverage.

Each of the sensor nodes act independently to collect and route data to other sensors or the base station (BS). The base station is usually an intelligent device with unlimited energy resources that is capable of connecting the sensor network to an external communication infrastructure or internet for easy usage of the reported data for decision making. It is also the point where aggregation, clustering and routing tasks are implemented for a dense WSN (high node density). Usually, sensor nodes can be deployed stationary or mobile over large areas. Though the sensor nodes can work autonomously, they work cooperatively to sense the physical conditions of an environment. Sensor nodes can sense the environment, communicate with neighbouring nodes, and in many cases perform basic computations on the data being collected [Akkaya and Younis \[2005\]](#); [Zungeru et al. \[2012\]](#). These attributes qualify WSNs to be an excellent choice for many applications [Yu et al. \[2006\]](#).

Sensors consist of four basic unit components: a sensing unit, a processing unit, a communication unit, and a power unit as shown in Figure 2.1. The sensing unit usually consists of sensor(s) and an analogue to digital converter bits (ADC). In sensing applications (such as weather monitoring, tactical surveillance, event detection etc), the sensor nodes sense or measure the physical condition of a monitored area. The ADC digitises a continuous analogue signal sensed by the sensors before sending it to the processing unit. This unit is made up of the memory-enabled micro-controller/microprocessor which provides the sensor nodes with intelligent control capabilities. The communication unit consists of a short-range radio capable of transmitting and receiving signal over a channel. The power unit is made up of a battery for supplying power that drive rest of the built-on system components [Al-Karaki and Kamal \[2004\]](#). However, one of the issues that sensors have is the limited energy supply of the battery and so there is need to employ energy conservation strategy in order to prolong the lifespan of sensors.

Routing is a major process to be considered in order to minimise the energy consumption in WSNs. Due to the limited transmission range of each node, it may be necessary for sensors to use other nodes to forward packets to the BS. Routes discovery and maintenance in WSNs is non-trivial due to the energy restrictions

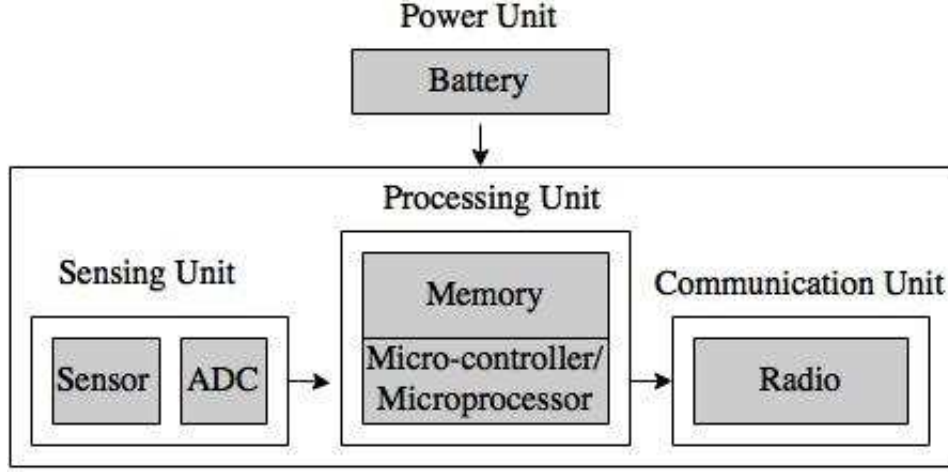


Figure 2.1: The components of Sensor node

and transmission range restrictions. To minimise energy consumption, routing protocols proposed in the literature for WSNs employ some well-known routing strategies such as clustering.

Clustering protocols in WSNs aim at grouping the sensors into clusters and selecting a cluster head (CH) for each cluster. In order to realise an energy efficient WSN, the CH can aggregate the data sent from the cluster members and send them directly to the BS. A clustering protocol is mainly a two layer protocol. The first layer deals with deciding the optimal CH set and the second layer protocol is responsible for transmitting the data to the BS. The clustering protocol in WSNs should not only facilitate data transmission, but also consider the sensor nodes' constraints. It should also meet the WSNs requirements including the energy efficiency, the data delivery reliability, and the scalability requirements (see section 2.1.2). Apart from clustering, the sleep scheduling mechanism is another energy saving technique that preserves the lifespan of sensors by causing sensor to sleep when not needed and awake or active intermittently. More details on sleep scheduling is covered in section 2.2.

2.1.1 Applications of WSNs

WSNs are used for different applications ranging from military to civil application such as medical, industry and home [Puccinelli and Haenggi \[2005\]](#). The various

2. Energy Efficiency Mechanisms for WSNs

applications can be categorised under the following sub-headings.

Environment Monitoring Systems

Environment monitoring systems are a crucial application that control and monitor environment conditions such as light, pressure, humidity and temperature [Othman and Shazali \[2012\]](#). Applications have grown rapidly for monitoring purposes such as indoor, greenhouse, agriculture, climate, habitat and forest monitoring. In [Chang et al. \[2012\]](#); [Lazarescu \[2013\]](#); [Nie et al. \[2014\]](#), several studies have been focused on this application aspect. The major WSN requirements of environment monitoring applications are **scalability, coverage and energy efficiency**. Apart from the previously mentioned requirements, there is need for data reliability in the case of indoor monitoring applications such as fire detection and alarm systems because it involves property and the protection of life. Monitored areas can span up to several square meters, so the number of nodes deployed over an area can vary from hundreds to thousands. Hence, scalability is a very important pre-requisite in the development of any protocols that can support very large quantities of nodes and guarantee full coverage of the monitored area [Rault et al. \[2014\]](#). The protocols proposed in this thesis are applicable for environment monitoring because they put into perspective the three essential design requirements; which are lifetime extension, coverage and reliability.

Human Body Monitoring

Research interest in the aspect of wireless health care systems has grown rapidly and contributed advantageously to increasing numbers of elderly people, ability to place patients under continuous health monitoring and the rising cost of medical services. The emergence of novel wireless human body monitoring system such as wireless body sensor networks (WBSNs) have unlocked the potential to a broad variety of assisted living applications such as biochemical/biophysical control, emotional recognition for social networking, activity monitoring for health care, e-fitness, emergency detection, security, and highly interactive games [Aiello et al. \[2011\]](#). Lots of efforts has been geared toward WBSNs for human body monitoring by researchers [Baskaran \[2012\]](#); [Gulcharan et al. \[2014\]](#); [Kateretse](#)

2. Energy Efficiency Mechanisms for WSNs

et al. [2013]. Human body monitoring is conducted using WSNs which is usually attached to the body surface and planted inside the human body tissue. Modern technologies have produced micro and intelligent medical sensors that can be worn or implanted into the body. The sensors extract data from different parts of the body systems and send it to a central device that performs aggregation and analysis. These applications require high reliability due to the involvement of human's life Souil and Bouabdallah [2011]. Another very important requirement that ensures lengthen period of system operation is the network's energy efficiency Baskaran [2012]; Souil and Bouabdallah [2011].

Intelligent Buildings

Intelligent and automated buildings is a WSN application that address increasing energy cost and aiding the green movement. Smart sensor nodes that can improve safety and security, minimise energy consumption and operational costs have been deployed for building automation applications. Using WSNs, Several literatures in Dounis [2010]; Fortino et al. [2012]; Suryadevara et al. [2015] have proposed several intelligent building management systems. In WSNs, different sensor types that measure parameters such as pressure, temperature, smoke and light are employed for intelligent building management systems. At different level and home appliances, this system may include servers, gateways, actuators, communication and application software Jaafar and Watfa [2013]. Intelligent building management systems require multi-hop communication approach for covering the whole building. Another vital system requirement for this purpose is the WSNs energy efficiency Jaafar and Watfa [2013]. Some hierarchical or data-centric protocol can be used to satisfy these requirement Fortino et al. [2012].

2.1.2 Requirements and Challenges of WSNs Design

WSNs consist of a large number of sensor nodes that are made up of miniature devices constrained in their stored energy capabilities. Therefore in order to increase their usefulness, energy efficiency is a pivotal system requirement in WSNs. WSNs should put into consideration the sensor nodes' short transmission range in the sending data to the sink. Data reliability is another core requirement

2. Energy Efficiency Mechanisms for WSNs

that must be considered in WSNs design. Clustering facilitates local interaction among sensor nodes in a coordinated manner which enables the WSNs to achieve the utmost goals such as scalability and efficient usage of limited energy resource [Tubaishat and Madria \[2003\]](#). Scalability simply means the system ability to work efficiently with improve performance as the network size increases [Lee et al. \[1998\]](#). Thousands of sensors are deployed in a large area of interest to compensate for the limited transmission range of each sensor. The most preferable routing scheme in WSNs is the one that can work efficiently with this large number of sensor nodes and must be capable of adapting to an increasing network size. Therefore, scalability is a another major requirement in the WSNs system design. To measure the performance of any clustering protocol in terms of scalability, the number of un-clustered is recorded as a performance metric. An increasing quantity of un-clustered nodes indicates a degrading performance in terms of the protocol scalability. WSNs researchers faces some challenges due to the unreliable nature of wireless communication and the limited resources of sensor nodes. The main challenges of the WSNs are listed as follows:

Limited Energy

Constrained energy supply poses a big challenge in the design of WSNs because sensors are powered on battery which has limited energy capacity. When the battery-energy of a sensor is depleted below a certain threshold, it becomes faulty and unable to work properly which can negatively affect the overall network performance. Due to the small size of sensors, batteries are usually designed in small sizes. Therefore, the overall operation of sensors is limited by the available battery energy. On the contrary, sensors need to continuously sense or collect, transmit and receive data for a long period of time. To ensure that sensors operate for long period of time, the battery energy must be managed efficiently. Consequently, the routing protocols adopted by sensor networks should achieves the energy efficiency requirement so as to minimise the energy consumption and hence extend the network's lifetime. Although, WSN applications faces different issues but the common challenge is the limited energy. Energy consumption is considered the main challenge for WSN operation. The sensor nodes are equipped

2. Energy Efficiency Mechanisms for WSNs

with limited batteries. They are deployed in hostile or unsafe areas; making recharging or replacing the battery unfeasible, hence the need to conserve battery energy [Raghunathan et al. \[2002\]](#).

Every node operation consumes energy. Energy is consumed in sensing, processing and communicating. Sources of energy consumption include:

- **Idle:** It reflects the time during which the node keeps listening to the channel waiting to receive data. The idle process consumes energy which can be considered as passive. The node could be designed to sleep during passive time and wake-up to receive data. Designing node's duty cycle to sleep and wake-up at the right time is still a challenge.
- **Data Aggregation:** Sending data messages from all sensors to the base station directly causes overheads due traffic congestion. Aggregating data can reduce communication traffic. This is done by combining data messages into one. Data aggregation requires the node to have sufficient memory, processor capabilities and energy for processing.
- **Communication:** Most of the node's energy is consumed during communication [Halkes et al. \[2005\]](#). The consumed energy during communication is affected exponentially by the distance between the communicating nodes; the longer the communication distance between sensors the more energy is consumed. In order to save energy, communication distance should be minimised. Moreover, designing a suitable pattern for the antenna help reducing energy waste. It was reported that the energy consumed for an antenna pattern to reach all hosts is proportional to the area it covered [Sravan et al. \[2007\]](#).

Short Transmission Range

Each sensor needs to transmit their data to the sink even though their transmission range is limited. The sink is normally fixed and located far away from the sensors. In addition, the link quality between the sensors nodes and the sink need to be enhance in order to facilitate network throughout and data reliability.

Clustering techniques that uses multi-hop communication approach is the best strategy that can satisfies this requirement.

Coverage Problem

Coverage can be defined as the extent or degree to which each grid point of network area is covered by a sensor node. The coverage problem is whether each or every point in the target or area of interest falls within the sensing range of the deployed sensors [Sangwan and Singh \[2015\]](#).

Scalability Issue

WSNs can consist of a large number of sensor nodes with high node density. Designers find it a great challenge designing a scalable protocol that can work efficiently at such a large network size.

Cluster and Route Optimisation Issues

In WSNs, there is a need to apply the best clusters that can route traffic in an energy efficient manner such that the overall energy consumption is minimised and the sensor lifetime is extended. The clusters formation and cluster head selection among the sensors is regarded as an optimisation problem that can be tackled using meta-heuristic algorithms (Refers to [Section 2.4](#)).

2.1.3 Energy Consumption in WSNs

As mentioned earlier, a wireless sensor node consists of: sensing unit, processing unit, transceiver and power supply. The power supply provides energy to all other sensor components. The sensed measurements are converted to a digital signal by means of the analogue-to-digital converter (ADC) of the sensing unit. The processing unit aggregates the digitised data into one single message to be sent by the transceiver. The total energy consumed by a wireless sensor is the amount of energy required to perform sensing, aggregation and transceiver operations. Typical operations of the transceiver are: sleep, idle, transmit and receive. The

2. Energy Efficiency Mechanisms for WSNs

total energy consumed is computed as:

$$E_{Total} = E_{SU} + E_{Aggreg} + E_{Sleep} + E_{Idle} + E_{Tx} + E_{Rx} \quad (2.1)$$

Where:

- E_{Total} is the total energy consumed by a wireless node,
- E_{SU} is the energy consumed by the sensing unit,
- E_{Aggreg} is the energy consumed in aggregating measured data,
- E_{Trans} is the total energy consumed by the transceiver,
- E_{Sleep} is the energy consumed by the transceiver during sleep operation,
- E_{Idle} is the energy consumed by the transceiver while in the idle state,
- E_{Tx} is the energy consumed by the transceiver to send a data message,
- E_{Rx} is the energy consumed by the transceiver to receive a data message.

Sensors spends most of its energy to transmit and receive packets whereas the energy consumed during the idle, sleep states and sensing unit is negligible (E_{Sleep} , E_{Idle} and E_{SU} is approximately zero). Equation 2.1 can be approximated to:

$$E_{Total} = E_{Aggreg} + E_{Tx} + E_{Rx} \quad (2.2)$$

2.2 Sleep Scheduling Mechanisms Overview

In WSN environments, sensor node sleep scheduling can be used as an energy conservation method for network lifetime extension. This section present few notable energy-efficient scheduling mechanisms in sensor networks.

Randomized Scheduling Scheme

In the randomized scheduling (RS) scheme, sleeping sensors are selected randomly in a given cluster with a probability of $p = \beta_s < 1$ (where β_s is the average fraction of sensors allowed to sleep). This scheme is very simple to implement. Each sensor only needs to examine the data obtained from a biased random generator to decide whether to turn into sleep mode or not. All sensors in the cluster have

same sleep probability. The major drawback with this scheme is that more energy is consumed if the sensors at cluster boundaries are kept active and this can also leads to variation of energy consumptions by all sensors. This sleep scheduling scheme is only suitable for single-hop cluster-based sensor networks with variable transmission power [Deng et al. \[2005a\]](#).

Linear Distance-based Scheduling

A sleep-scheduling algorithm called Linear Distance-based Scheduling (LDS) scheme for cluster-based high density sensor networks was proposed in [Ramesh et al. \[2012\]](#). The goal is to reduce energy consumption without affecting the coverage capabilities of the sensors. This goal is satisfied by causing sensors that are far away from the cluster head to become inactive with higher probabilities. Also, experimental result shows that the LDS scheme is better than the RS in that the sensing coverage of sensors at the border area are lower than the central area of the cluster [Deng et al. \[2005b\]](#). The idea behind this scheme is that more sensor energy can be saved by allowing far away sensors to sleep for longer periods compared with sensors that are closer to the cluster head. According to [Deng et al. \[2005a\]](#), the probability p a sensor goes into sleep mode is given as:

$$p(x) = \frac{2R\beta_s}{4} \frac{2x}{R^2} = \frac{3\beta_s x}{2R}, \quad 0 \leq x \leq R \quad (2.3)$$

Where R is the communication range between CH and all sensors at maximum transmission power, β_s is the fraction of sensors allowed to sleep (at $< 2/3$). The sleep probability p of sensors is a function that is dependent on x , which is the distance of a sensor from its respective cluster head and x is a value within sensor's communication range R . The LDS scheme works only with static clusters (CHs remain the same throughout an operation once they are selected). The LDS scheme lowers the variation of energy consumptions compared with the RS scheme [Deng et al. \[2005a\]](#).

Balanced-energy Sleep Scheduling

This scheme employs the base station to extend the LDS scheme by evenly distributing the sensing and communication tasks among the non-head sensors so that their energy consumption is similar regardless of their distance to the cluster-head [Le et al. \[2008\]](#). This scheme employs a sleep probability function $p(x)$ so that the total energy consumption of a sensor does not depend on x , the distance between sensor and its CH. The main goal of this scheme is to use a sleep probability that provides balanced energy consumption for a larger portion of sensors in a cluster thereby reducing the overall energy consumption. More details about this scheme is provided in [Deng et al. \[2005a\]](#)

Other Sleep Scheduling Mechanisms

In [Danratchadakorn and Pornavalai \[2015\]](#), a coverage maximisation with sleep scheduling protocol (CMSS) that ensures network areas are fully covered by selected active sensors was presented. Each sensor exchanges information with its neighbouring sensors and sets a waiting time. During sensor waiting times, a sensor can receive a sleep message from neighbouring nodes. When a sensor receives these messages, it updates its own neighbour and cell value table. If the minimum value of the cell value table of a sensor equals to one, it silently becomes an active node. Otherwise, it will wait for the waiting time to expire before it turns into an inactive node. An energy preserving sleep scheduling (EPSS) strategy allows each sensor to make decision regarding going into sleep mode based on their distance from the cluster head and network density. This guarantees balanced energy consumption in the cluster by taking into account the density of node deployment and the network load while determining the sleep probability [Singh and Lobiyal \[2013\]](#). In [Bulut and Korpeoglu \[2011\]](#), a probabilistic and analytical method was employed to approximate the overlapping sensing coverage between a node and its neighbours. It also estimates when a node can be put into sleep without jeopardizing expected coverage. The method is employed by the proposed scheduling and routing scheme to diminish control message overhead while considering the next mode (full-active, semi-active, inactive/sleeping) of sensor nodes.

2.3 Clustering Mechanisms Overview

Clustering is a key mechanism in large multi-hop wireless sensor networks for obtaining scalability, reducing energy consumption and increase in the network lifetime to achieve improved network performance. In [Abbasi and Younis \[2007\]](#); [Tyagi and Kumar \[2013\]](#); [Younis et al. \[2006\]](#), Clustering techniques have been studied extensively to improve the performance of WSNs. In this section, several traditional clustering protocols were presented, which includes the Low Energy Adaptive Clustering Hierarchy (LEACH), Topology Controlled Adaptive Clustering (TCAC), Scalable Energy Efficient Clustering Hierarchy (SEECH) and other clustering approaches.

2.3.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH is one of the most common cluster-based routing protocols in WSNs that has been proven to be an effective approach to prolong the network's lifetime [Heinzelman et al. \[2002, 2000\]](#); [Tyagi and Kumar \[2013\]](#). The LEACH protocol was published in a seminal paper by [Heinzelman et al. \[2002\]](#), and has been cited in most research papers of similar research area. This is a completely distributed approach that does not require a global information of the network. The basic idea of LEACH has been an inspiration for many subsequent clustering protocols. The main objective of LEACH is to equalise the energy load distribution among the CHs. LEACH lifetime operations is made up of several rounds and each round consists of two phases, namely the set-up phase and the steady-state phase. In the set-up phase, the clusters are organised, while in the steady-state phase, data is delivered to the BS. The steady-phase span through longer period compared with the set-up phase in order to reduce overhead. For each, the node decide whether to be the CH or not at the set-up phase. This CH decision is based on the percentage allocation of CHs or the number of times the sensor has been a CH. Cluster-heads can be chosen stochastically (randomly based). At the set-up phase of each round, a stochastic threshold value $T(n)$ is computed at each round

2. Energy Efficiency Mechanisms for WSNs

as defined below:

$$T(n) = \begin{cases} \frac{P}{1-P \times (r \bmod \frac{1}{P})}, & \text{if } \forall n \in G \\ 0, & \text{Otherwise} \end{cases} \quad (2.4)$$

Where n is a random number between 0 and 1, P is the desired CHs percentage, r is the current round, and G is the set of nodes that have not been elected as CHs in the last $\frac{1}{P}$ rounds. During set-up phase, each sensor nodes will select a random number n between 0 and 1. This random number n is compared with $T(n)$ and if it satisfies the condition $n < T(n)$, the node becomes a CH. This LEACH protocol ensures that every node becomes a CH exactly once within $\frac{1}{P}$ rounds.

When a cluster head role is assigned to a node and this node announces its new role to other nodes via an advertisement message. All the nodes decide which CH to join based on the received signal strength of the advertisement message. Each node responds to their respective CH via a membership message. Using equation 2.4, the CH role is decided in order to distribute the energy load among sensors. During the steady-state phase, the sensors transmit data packets continuously to the CHs. Each CH aggregate all the data received from all its member nodes and this aggregated data is sent to the BS directly. To avoid inter-cluster and intra-cluster collisions, LEACH employs Time Division Multiple Access (TDMA) technique. After a time duration or round length, the WSNs begins another round starting with the set-up phase where new CHs are elected.

The idea behind LEACH is that any node that has been appointed as CH in a round can not be elected as CH again. This LEACH scheme enables each node to share equally the extra energy load imposed by acting as the CH. However, the drawback is that LEACH can not guarantee equal load-balancing in the sense that sensors are elected as CHs based on probabilities without considering their energy value while choosing $T(n)$. In addition, selecting the CHs randomly does not gives an even distribution of CHs over the WSNs [Arboleda and Nasser \[2006\]](#). Another drawback is that LEACH assumes a single-hop communication with the BS, which is unrealistic in many practical scenarios due to the restricted communication range of sensors [Saleem et al. \[2011\]](#); [Zungeru et al. \[2012\]](#). An

attempt to increase the transmission range can also cause sensors to consume more energy for transmission.

2.3.2 Topology Controlled Adaptive Clustering (TCAC)

The TCAC protocol consists of three phases namely the periodical update, cluster heads election and cluster formation [Dahnil et al. \[2012\]](#). At every network operation round, each sensor dynamically changes its transmission power level P_{tx} . At the start of periodic update, each sensor successively broadcasts an update message containing its ID, at each power level. Other sensors will send an acknowledgement (Ack) after receiving the packets. Sensor's degree is computed based on the number of received Acks. The broadcasting sensors must perform many transmissions to obtain a power level that is equivalent to the degree threshold (Q_{min}). To preserve the network connectivity at a given number of sensors n , the degree threshold Q_{min} is defined as:

$$Q_{min} = 5.1774 \log n \quad (2.5)$$

According to [Xue and Kumar \[2004\]](#), the relevance of Equation 2.5 is that as each sensor is connected to more than $5.1774 \log n$ nearest neighbour, the network is asymptotically connected with a probability approaching one as n increases. For instance, if a sensor's degree is less than the degree threshold Q_{min} , then the sensor must increase its power level. Alternatively, if the sensor's degree is greater than the threshold, then the sensor must reduce its power level until Q_{min} is achieved. This attainable power level by the broadcasting sensors is set as the base power level. This information will be stored in the sensor's cache for clustering operations. The cluster heads are elected in the second phase of TCAC protocol in three sequential steps. In Step 1, a sensor computes its probability ($P(CCH_i)$) to become CH candidate based on its remaining energy E_i with respect to the average energy E_{avg} of all sensors in the WSNs. LEACH protocol defines the optimal number of cluster heads k_{opt} that can achieve minimum energy dissipation per round. A parameter $k_{initial}$ that can obtain non-overlapped CHs is defined in TCAC protocol and this parameter must satisfies the condition $k_{initial} > k_{opt}$.

2. Energy Efficiency Mechanisms for WSNs

The probability of a sensor to be elected as a CH candidate is computed as follows

$$P(CCH_i) = \frac{E_i}{E_{avg}} \quad (2.6)$$

And E_{avg} is calculated as follows

$$E_{avg} = \frac{E_{total}}{k_{initial}} \quad (2.7)$$

Each sensor generates a random number in the range of $[0, 1]$ and if the number is less than the calculated probability $P(CCH_i)$, it elects itself as a candidate CH. Other sensors that are not elected will wait to receive membership message from the newly elected CH. At the end of each round, all member sensors send their residual energy to their respective CH, which aggregate the values and update the rest of CHs in the network. Based on the received information, the CHs compute E_{avg} value. The information is sent to all sensors in order to compute the $P(CCH_i)$ for the next round of network operation. There is a higher chance that sensors elected as CH in previous round may not be elected in the next round due to higher energy spent for communication with the base station. In Step 2, the candidate CHs obtained in Step 1 is compared against each other and the condition below must be satisfies for CH election.

- The CH candidate with a higher energy is re-elected as a CH and the other candidates becomes a member nodes.
- If the energy of two candidate CHs is the same, then the candidate CH with higher degree is re-elected as the CH.

In Step 3, CHs set power level P_{tx} through Ack counts received from transmission in order to update the sensor's degree. If the sensor degree correspond to the Q_{min} , the CH transmit a cluster head message *CHMSG* its new role. In the cluster formation phase, the non-CHs respond back to CH with a request message *REQMSG* containing its ID. Sensors that do not receive *CHMSG* across the network send request message to CHs after the time has expires. CH rank members based on the received signal strength of *REQMSG* and stored them in a priority list in the structure (ID, Rank). At the top of priority list is the

sensor with strongest signal strength. CH broadcast the priority list to all sensors requesting to join its cluster. The two important facts can be deduce from the priority list:

1. Sensors are aware about their closeness to the CH compare to other sensors requesting to join the cluster,
2. Sensor may join cluster to fulfil the threshold degree of CH rather than considering the closeness to CH. From the list, non-CH nodes compute the degree of each CH and a sensor joins cluster that has a lower degree than the Q_{min} .

In situation where the degree of all cluster heads are equal or greater than the Q_{min} , the sensors compare ranks given by all CHs and join the cluster that placed it in the best rank. After joining the cluster, member sensors adjust their transmission power for efficient intra-communication with CHs.

2.3.3 Scalable Energy Efficient Clustering Hierarchy (SEECH)

After sensor deployment in a network area, The SEECH protocol starts operation at the start phase before the first round. Each sensor computes its distance from the sink and the number of neighbouring sensors n_i in a specific radius RNG [Tarhani et al. \[2014\]](#). This obtained data is shared with other sensors and each sensor computes its degree deg_i as follow

$$deg_i = \frac{n_i}{\max(n_1, n_2, \dots, n_N)} \quad (2.8)$$

Where n_N is the overall number of sensors in the network. In SEECH protocol, sensors with larger degrees are more suitable choices for cluster head. The merit of this approach is that large number of member nodes is covered by small number of CHs using low power communication. The CH selection starts by electing some tentative CH using distributive method. In this method, each node i calculates

2. Energy Efficiency Mechanisms for WSNs

p_{c_i} which describes the chances of being a tentative CH as follows:

$$p_{c_i} = \begin{cases} \frac{E_{res_i} \times deg_i}{p_{c-tot}}, & \text{if } E_{res_i} \geq E_{av}(1 - \lambda) \\ 0, & \text{else} \end{cases} \quad (2.9)$$

Where the E_{res_i} denotes the residual or available energy of sensor i . λ is a number in the range $[0, 1]$ and is usually set to 0.9 in order not to reduce the chances of low energy sensors. p_{c-tot} is define as follows:

$$p_{c-tot} = \frac{E_{av} \times \sum_N deg_i}{2K_{CHC}} \quad (2.10)$$

The p_{c-tot} value assures that the number of tentative CHs will not be less than the number of needed candidates (K_{CHC}). In equation 2.10, E_{av} is the average residual energy of the nodes in the current round which is calculated and broadcasted by cluster heads in previous round and during cluster formation. In each round, residual energy of nodes is in a small range; whereas, their degrees might be completely different e.g. degree of a node might be multiple times of another one. As a result, prioritizing the nodes by equation 2.9 is more heavily dependent on node degrees rather than residual energy. When p_{c_i} is computed, each sensor generates a random number in the range $[0, 1]$ and compares it with p_{c_i} . If the random number is less than the p_{c_i} , the sensor consider itself as the tentative cluster head. All tentative CHs inform other tentative CHs and sensors by broadcasting a CH-CANDIDATE-MSG message using CDMA protocol. Each sensor receive the message and estimate its distance from the transmitting node.

Prior to announcing its candidacy, each sensor counts the number of candidates and if it is equal to K_{CHC} , it will not introduce itself as a candidate to the network and gives up the competition in order to maintain constant number of candidates. Each *CH-CANDIDATE-MSG* message which includes the ID, residual energy of sender and distances smaller than RNG from all previously introduced candidates in current round and corresponding IDs. The set of candidates and cluster heads is denoted by R_{CHC} and R_{CH} and K_{CH} is the number of needed cluster heads. To eliminate each candidate, all candidates are scored once. The candidate with lowest score will be eliminated from the list. The same procedure will be repeated ignoring distance from eliminated candidate. The pro-

2. Energy Efficiency Mechanisms for WSNs

cess will continue till the number of remained candidates equals to the number of needed cluster heads. The SEECH protocol can execute algorithm in two ways. In the first method one candidate executes desired algorithm and announces final cluster heads. In the second method all candidates separately execute the algorithm and figure out their state themselves.

A sensor accepts the role of relay node only if it satisfies two conditions. Firstly, the sensors that are closer to the sink minimise transmission cost. Protocols must avoid selecting this sensor as cluster head. Secondly, since competence of electing a CH is proportional to node degree (which is number between 0 and 1), 1-degree is utilised and defined as the relay sensors. The procedure of selecting relay nodes is similar to selecting cluster heads. First of all, each node excluding cluster heads, calculates its chance of becoming a tentative relay node, p_{ri} , as follows:

$$p_{c_i} = \begin{cases} \frac{E_{res_i} \times (1 - deg_i)}{p_{r-tot}}, & \text{if } E_{res_i} \geq E_{av}(1 - \lambda) \\ 0, & \text{else} \end{cases} \quad (2.11)$$

Where

$$p_{r-tot} = \frac{E_{av} \times \sum_N (1 - deg_i)}{2K_R} \quad (2.12)$$

The p_{r-tot} value assures that the number of tentative relay sensors will not be less than the sufficient number of tentative relay sensors (K_R). In Equation 2.11, E_{res_i} is included to protect low energy nodes. When pr_i is computed, each sensor generates a random number and compares with pr_i . If the number is smaller then the sensor becomes a tentative relay sensor. The relay sensor introduce themselves to the network by broadcasting a *RLY-MSG* message which consist of the node ID and residual energy. CHs also receive the message and decide the closest relays based on the signal strength. After the lowest energy nodes has been eliminated, the CHs chooses closest relay nodes among the tentatives and informed the elected relay about their choice by sending *CH-NEXTHOP-MSG* message.

Cluster formation process starts by CHs broadcasting a *CHMSG* message with the spreading code and ID. Each normal sensor chooses the closest CH

according to the signal strength. Afterwards the sensor informs the CH of its decision by transmitting a *JOIN-MSG* message which consists of node ID and cluster head ID, thereby forming clusters. CHs count their members based on the number of received *JOIN-MSG* messages and employ TDMA to collect cluster information. Therefore, each CH broadcast a *SCHEDULE-MSG* message with a radius equivalent to the distance of farthest sensor member. Using the messages CHs issue time-slots for each members to send its information. Also the average residual energy of the network is informed to the nodes by these messages. Now setup phase finishes and steady-state phase starts in accordance with determined topology.

2.3.4 Other Clustering Approaches

Another variant of LEACH protocol was proposed in [Heinzelman et al. \[2002\]](#), which is called **LEACH-centralised (LEACH-C)**. Unlike LEACH, LEACH-C employs the sink to perform the task of CH selection and formation. Each node sends their location and energy level to the sink. The sink employs a simulated annealing (SA) approach to determine the CH number and cluster configuration based on the received information. The energy and distance between CHs and non-CHs are considered for even load and cluster distribution. The sink optimises global knowledge of the network to produce an improved network that requires less energy. However, it assumes that the CHs can send aggregated data streams directly to the sink which is a similar drawback to LEACH. A **hybrid energy efficient distributive (HEED)** protocol was proposed in [Younis and Fahmy \[2004\]](#). CH selection is achieved by iteratively considering the residual energy and the proximity to member nodes. In this protocol, the energy consumption for communicating between the CHs and non-CHs is reduced considerably and each CH communicates with the sink using multi-communication approach. However, more CHs are generated than the expected number and this results in an unbalanced energy consumption. Also, HEED results into overhead since it does several iteration to select CHs.

In [Smaragdakis et al. \[2004\]](#), a **stable election protocol (SEP)** was developed for the two level heterogeneous networks, which includes two types of nodes,

normal and advanced nodes. In SEP election probabilities are weighted by the initial energy of a node relative to that of other nodes in the network. This prolongs the time interval of FND that must be crucial reliable communication. Further SEP is dynamic therefore it does not assume any prior distribution of the different levels of energy in sensor nodes. Finally SEP is scalable as it does not require any knowledge of the exact position of each node in the field. Disadvantage of SEP is that it performs poorly in terms of stability for multi-level heterogeneous WSNs. An energy-aware adaptive clustering protocol used in heterogeneous wireless sensor networks named **distributed energy-efficient clustering (DEEC)** scheme was proposed in [Qing et al. \[2006\]](#). In DEEC, every sensor node independently elects itself as a cluster-head based on its initial energy and residual energy. To control the energy expenditure of nodes by means of adaptive approach, DEEC use the average energy of the network as the reference energy. Thus, DEEC does not require any global knowledge of energy at every election round. Unlike SEP and LEACH, DEEC can perform well in multi-level heterogeneous wireless sensor networks as shown in experimental results presented in [Qing et al. \[2006\]](#).

2.4 Metaheuristic Algorithms

Meta-heuristic approaches are widely employed as an efficient solution for many optimisation problems. They are defined as a heuristic process that intelligently combines diverse concepts in order to exploit and explore the search space, learn strategies that are tailored for finding solution close to the optimal solution [El Emary and Ramakrishnan \[2013\]](#). The meta-heuristics algorithm is classified into two types based on the search strategy namely the global and local meta-heuristic search. Here, a brief overview of widely known global search meta-heuristics is presented: Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), Differential Evolution and the Local search metaheuristics: Simulated annealing (SA), Iterated Local Search (ILS), Tabu search. Lastly, the Memetic algorithm (MA) , which is a hybrid approach of the global and local search strategy.

2.4.1 Global Search Strategy

The global search strategies are also referred to as the population-based metaheuristics. Population-based approaches maintain and improve multiple candidate solutions, often using population characteristics to guide the search. The global search provides the broad exploration mechanism. Algorithms under the population based metaheuristics include Genetic Algorithm, Particle Swarm Optimisation and Differential Evolution.

Genetic Algorithms

Genetic algorithms have been in existence earlier than 1975, but it was introduced to the broad research community in a seminal paper by Holland [1975]. The genetic algorithm concept is a metaheuristic that imitates the process of natural selection and the survival of the fittest. In GA, solutions are represented as chromosomes. Using a fitness function, the quality of these chromosomes is evaluated and eventually graded from the best to worst based on the obtained fitness value. To produce a high quality solution for optimisation and search problems, the GA mimics three major natural selection operations of living organisms such as selection, crossover, and mutation. To drive the GA process towards survival of the fittest, higher selection probability is assigned to chromosomes with better fitness. The selection probabilities are computed by ranking the fitness values of all chromosomes in a population relative to one another. The selection operator is applied to a population pool in order to select a parent chromosome pair with the best fitness value. Afterwards, the crossover operator is applied to this pair, thereby producing a new offspring. Since the process is continuously driven towards the stronger (fitter) chromosomes, there is a likelihood that the fitness of new chromosomes might tend towards the same value after several generations. Invariably, this causes a decline in the population diversity, which leads to population convergence. At this stage, the mutation operator can be applied to the process and this introduces diversity into the population to stop convergence Gen et al. [2008]; Goldberg [1989]. The flowchart of genetic algorithm is given in Figure 2.2.

The population size and the maximum number of iterations are among the

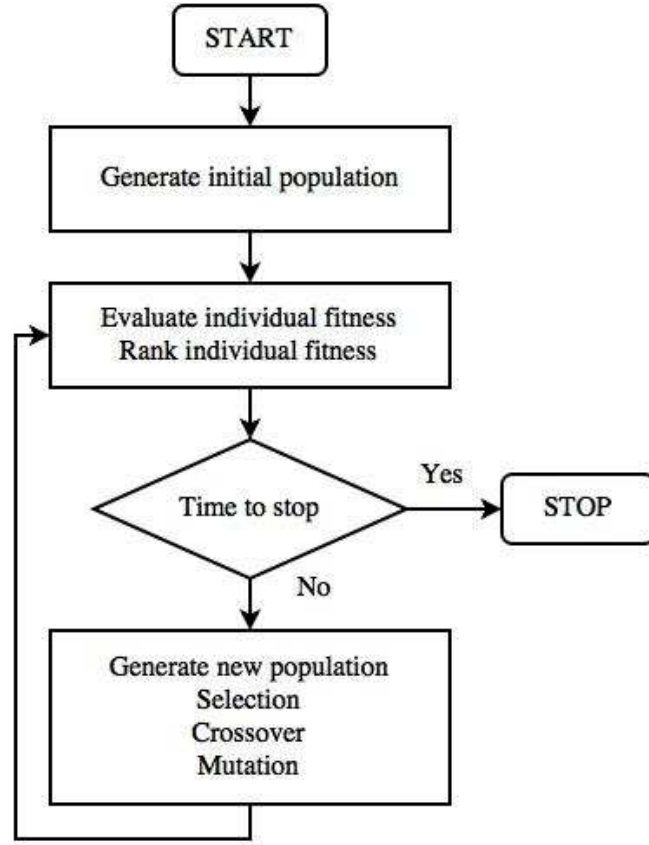


Figure 2.2: Flowchart for genetic algorithm [Kachitvichyanukul \[2012\]](#)

decisions that must be made for implementing a GA. The first decision made is the selection techniques and probability assignment mechanism, which makes use of the fitness values. The choices of appropriate selection techniques with probability assignments mechanisms is a paramount for obtaining new population diversity and solution improvement. Many selection techniques have been proposed in literature but the two popularly used mechanisms are the tournament and roulette wheel selection mechanisms. The crossover method and crossover probability is the second decision set made for producing new chromosomes. At the early stage, the simple crossover was employed but it tends to produce chromosomes that are unfit for many complex optimisation problems and this has leads to discovery of other preferable crossover which are one-point, two-point, uniform and so on. The last decision set is the mutation method and mutation probability, which

ensure population diversity by inserting new features into the chromosomes.

Particle Swarm Optimisation

Kennedy and Eberhart [1995] presented the PSO technique at the Congress on Evolutionary Computation and this triggered the series of publications on the successful application of PSO as a viable solution to complex optimisation problems. This population-based optimisation technique was inspired by the social behaviour of bird flocking for conceptual visualisation of search process. PSO is a metaheuristic as it makes few or no assumptions about the problem being optimised and can search very large spaces of candidate solutions. However,

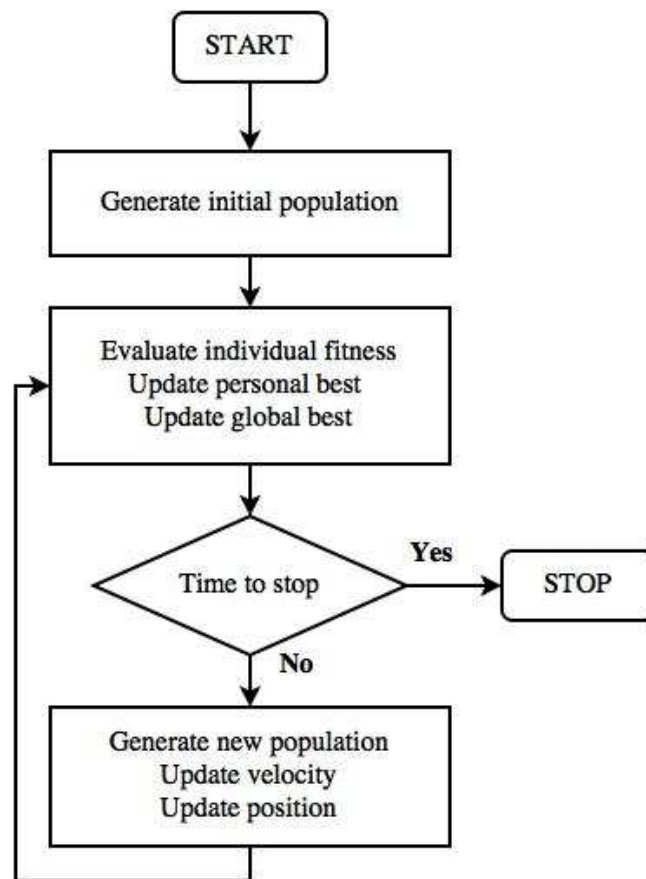


Figure 2.3: Flowchart for Particle Swarm Optimisation algorithm
Kachitvichyanukul [2012]

2. Energy Efficiency Mechanisms for WSNs

metaheuristics such as PSO do not guarantee an optimal solution is ever found. In PSO, a single solution is referred to as the particle and population is seen as the swarm. Each particle possess two features namely the position and velocity. Each particle uses the velocity to move to a new position. Once the new position is reached, the best position of each particle and the best position of the swarm is updated. Based on the experiences of the particle, velocity of each particle is then adjusted. The cycle continues until the stopping criterion is met as shown in Figure 2.3.

With GA, the first procedure is to generate an initial swarm and each particle in the swarm is initialised with a random position and velocity. The concept of solution is represented in the same manner as the GA. The fitness of each particle is evaluated using a defined fitness function. Every time a new particle is produced, the fitness is calculated and compared with the previous best particle. The personal and global best positions are updated to create a new swarm until the stopping criterion is satisfies. The velocity is updated by using the personal best, global best positions and old velocity. In another sense, the velocity is updated using an individual particle experience (personal or self learning term), swarm experience (group or social learning term) and old velocity. A weight is assigned to each term. Their is no need for fitness ranking and this makes it to undergoes less computational tasks than the GA. A simple arithmetic operation of real numbers is require for the velocity and position updates.

Differential Evolution

DE was proposed the same year with PSO by [Storn and Price \[1995\]](#) as an optimisation technique applied over continuous search space. The logic behind its operation is very simple and shows good performance in terms of convergence. DE converges at a slower rate towards the local optimum than the PSO but it has been a favourable optimisation technique applied to some applications [Onwubolu and Davendra \[2006\]](#). In DE, the solution represented in a vector of D-dimensional. The DE initialisation process starts by generating a random population size N of D-dimensional vectors which contain real numbers. The solution representation in DE is similar to GA and PSO. The DE algorithm uses a new mechanism for

2. Energy Efficiency Mechanisms for WSNs

generating new solution that differs from GA and PSO. In DE, several solutions is combine with the candidate solution to produce a new solution. DE algorithm simultaneously employs three main operators namely mutation, crossover and selection. These operators does not function exactly the same way as the one described in GA. The major process in DE is the generation of a trial vector by using a target or candidate vector from D-dimensional vectors of population size N. This process is achieved by using the mutation and crossover operator, which is summarised as follows:

1. Generate mutant vector by mutating three randomly selected vector.
2. Generate trial vector by applying a crossover on the mutant and target vector.

The first procedure requires randomly selecting three vectors from a population of vectors with the exception of target vector. The mutation operator is applied on these three randomly selected vectors X_1, X_2 , and X_3 and combined to obtain the mutant vector V as described in the Equation 2.13 below as:

$$V = X_1 + F(X_2 - X_3) \quad (2.13)$$

Where F is a multiplier which is the key parameter of the DE algorithm. The concept of mutation operation used in DE is completely different from GA. The second procedure is to generate a trial vector by using a crossover between the target and mutant vector. The two popularly employed crossover methods are the binomial and exponential crossover. The value of crossover probability determines the trial vectors in the sense a higher value causes the trial vector to be close to the mutant vector, whereas a small value causes the trial vector to resemble the target vector. After the formation of trial vector for a given target vector, both vectors are compared and one is selected. The selection criterion is based on the best fitness value; which means if the trial vector fitness is better than the target vector, the trial vector is selected. On the other hand, if the target vector has a superior fitness than the trial vector, the target vector is selected to be crossed over with the next round of mutant vector. This is a significant difference in the sense that solution improvement may occurs when waiting for

the entire population to finish the update. The flowchart for the DE algorithm is presented in Figure 2.4.

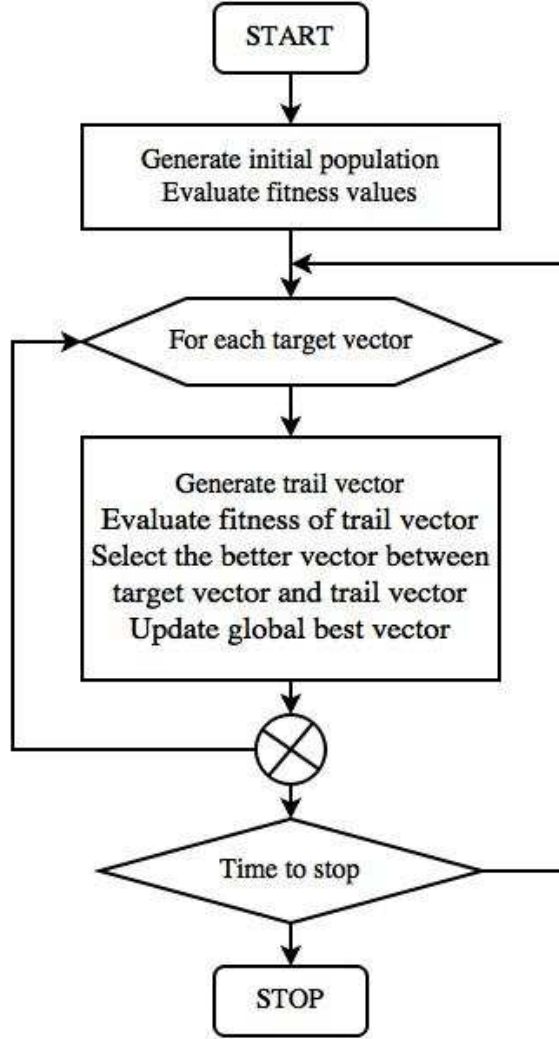


Figure 2.4: Flowchart for differential evolution [Kachitvichyanukul \[2012\]](#)

As presented in Figure 2.4, the first step is the generation of solutions in a population called vectors. The fitness of each vector is evaluated using a defined fitness function. A trial vector is generated for each target vector in successive order. The process coordinates the selection of either the target vector or the trail vector based on their fitness value. Eventually, the vector that wins between the trial and target vector moves to the next round and the one that loses is discarded. The observations drawn from the DE process are that new solutions

emerge only if the trial vectors has a better fitness. The overall average fitness of the population improves or remain constant from iteration to iteration. Solution that has undergone improvement in previous round are readily available to produce mutant vector for the next target vector. The DE differs from the GA in the sense that solution improvement occurs after all the solutions has finished their iterations.

In DE, every solution in a population are eligible to be use as a target vector or one of the parents unlike GA where the parent solution are selected based on fitness. The second parent is formed from at least three different vectors. In total, four vectors are used to generate the trial vector; and this trial vector replaces the target vector only if it has a higher fitness value otherwise it is discarded. This replacement takes effect immediately without going through the entire population to complete its iteration process. In the next round, this improved vector will then be available for crossover operation with the next mutant vector. Other variation of DE discussed in [Price et al. \[2006\]](#) lies in the formation process of mutant vectors and the use of more than three vectors for mutant vector formation.

2.4.2 Local Search Strategy

The local search strategy can also be called a single solution approach. This approach focus on modifying and improving a single candidate solution. The improvement of an individual solution via a local search strategy implies an exploitation mechanism. Popular algorithms under this approach includes the Simulated annealing (SA), Iterated local search (ILS) and Tabu search (TS).

Simulated Annealing

Simulated annealing (SA) is a meta-heuristic approach is employed as an approximate method in global optimisation for exploiting a relatively large search space. The logic behind its operations lies the process of metal annealing [Tan et al. \[2011\]](#). One major advantage of SA over gradient-based methods and other deterministic search methods is ability to avoid being stuck at the local optimal. A local optimal can be describe as dropping some bouncing balls over a landscape, and the balls bounce until they loses energy and settle in some local

2. Energy Efficiency Mechanisms for WSNs

minima location. Also, the balls can be allowed to bounce for a long time and loses energy slowly till they settle in a global minima location. SA searches along the Markov Chain until it converges under an ideal conditions.

In SA, the search is moves through a piecewise path. An acceptance probability is evaluated for each move and it accepts a move only if there is improvement of the fitness function (lower fitness value for minimisation problem or higher fitness value for maximisation problem), but also keeps record of changes that does not improve the fitness function. The acceptance probability p is computed as:

$$p = \exp \left[- \frac{\Delta E}{k_B T} \right], \quad (2.14)$$

Where k_B is the Boltzmann's constant, T is the temperature for controlling the annealing process and ΔE is the energy change. The variation in fitness function Δf is proportional to the energy change ΔE as shown below.

$$\Delta E = \gamma \Delta f, \quad (2.15)$$

Where γ is a real constant that is simply taken to be $\gamma = 1$ and $\Delta f = f_2 - f_1$ (f_1 and f_2 represent the old and new fitness function respectively). From equation 2.14, it is clear that $p \rightarrow 0$ as $T \rightarrow 0$, which makes the SA approach to implicitly behave like a hill-climbing method at extremely low temperature. In search algorithm, the acceptance probability function takes in the variation between the new old and new fitness function Δf and current temperature T , and produces a p value that decides whether to move to the new solution. The annealing Equation 2.14 help the search process to move from a random solution to one with a very good fitness function. This equation means that the acceptance probability:

- is always > 1 when the new solution is better than the old one. p is usually assumed to be 1 for a probability greater than 100%,
- gets smaller as the new solution gets more worse than the old one,
- gets smaller as the temperature decreases (if the new solution is worse than the old one).

2. Energy Efficiency Mechanisms for WSNs

The control of temperature variations is responsible for behaviour and efficiency of the SA algorithm. There are several ways to control the temperature decrease or cooling rate. Two popularly used annealing or cooling schedules are the linear and geometric. Equation 2.16 is applicable to linear cooling schedule.

$$T = T_0 - \beta t, \quad (2.16)$$

Where T_0 is the initial temperature, t is a pseudo time that replaces the iterations and β is the cooling rate. β value is decided such that $T \rightarrow T_f$ when $t \rightarrow t_f$ (or maximum number of iterations), thereby resulting into $\beta = (T_0 - T_f)/t_f$. On the other hand, a geometric cooling schedule essentially decreases the temperature by a cooling factor $0 < \alpha < 1$ so that T is replaced by αT or

$$T = T_0 \alpha^t, \quad t = 1, 2, \dots, t_f. \quad (2.17)$$

The merit of geometric cooling schedule is that the maximum of iterations is irrelevant as the $T \rightarrow 0$ when $t \rightarrow \infty$ and this makes the method to be more widely applied. The cooling process should be slow enough so that the system can attain stability. In practice, $\alpha \in [0.7, 0.99]$ is commonly applied. At a given temperature, multiple evaluations of the fitness function are used. Very few evaluation might cause pose an instability threat and the system might eventually converge into its global optimum. On the other hand, Too many evaluations consumes lots of time and converge slowly; the number of iterations increases exponentially before it reach stability depending the problem size. Hence, there is a trade-off between the number of evaluations and the solution quality. Evaluation can only be perform at few temperature levels or few evaluations at many temperature levels. The number of iterations can be either set at fixed or variable.

Iterated Local Search

Iterated local search (ILS) is the modification of hill climbing method for solving discrete optimisation problem. This search method can get trapped in the local optimal where no improving neighbours are available. ILS is a simple method that starts its search from a solution and apply the local search with perturbation operator to obtain the local optimal solution \hat{s} . This steps are iteratively repeated

Algorithm 1 Pseudocode of the ILS method

```

Start
   $s \leftarrow$  generate initial solution.
  Repeat
     $s' \leftarrow$  perturbation ( $s$ )
     $\hat{s}' \leftarrow$  local search ( $s'$ );
     $\hat{s} \leftarrow$  apply acceptance criterion ( $\hat{s}', \hat{s}$ );
  Until termination condition
End

```

until it satisfies its stopping criterion. Figure 1 shows the pseudo-code of ILS approach. The initial solution should contain detailed information so that it can serve as a starting point for local search. Most local search operators are deterministic. The perturbation operator is component added to the local search operators to make it non-deterministic and introduces exploration mechanism in the search for solution. Perturbation operator does a global random search around the regions of local optimum. An efficient use of the perturbation mechanism is a major requirement is discussed in Xu et al. [2003]. The perturbation must be strong enough to escape from spot of attraction, but less strong to use knowledge from previous iterations. Less knowledge from previous iterations, makes the ILS to be different from other restart strategy. The acceptance criterion is only satisfies if there is improvement in the solution. Section 5.4.2 of Chapter 5 discussed more details on this local search operator.

Tabu Search

Tabu Search was developed by Fred Glover as a search that employ memory and search history as a key component of the technique Glover [1995]. Previously, the use of search history was not seen to be important as most algorithms are memoryless or use the most current results. The goal of Tabu Search is to constrain an embedded heuristic from revisiting an area in recently discovered search space. The strategy maintain a short term memory that save recent changes to solutions within the search space while denying future moves from undoing those changes. An intermediate-memory can be introduced to allow more bias moves towards viable areas of the search spaces, as well as longer-term memory structures that

promote a general diversity in the search across the search space. The action of memory and history in the search procedure introduces a high degree of freedom and the mathematical analysis of the algorithm behaviour can be difficult to handle. However, TS is considered to be one of the best meta heuristic techniques use for optimisation problems.

TS can be seen as an deep local search procedure that efficiently employs the use of search history to avoids the search twice for a local solution that has been tried and recorded in the tabu lists. The search efficiency can be improved by recording the all tried solutions in the tabu list over a large number of iterations and this helps to save lots of computing time and resources. Wang and Nie [2010] shows that use of tabu lists with integer programming can save the computing time by at least two orders of magnitude for a given problem compared with traditional integer programming without tabu list. Several hybrid algorithms have been proposed by combining Tabu Search with other meta heuristics algorithm.

In essence, TS can be considered as an intensive local search, and the appropriate use of search history avoids revisiting local solutions by recording recently tried solutions in tabu lists. Over a large number of iterations, these tabu lists could save a significant amount of computing time, leading to improvements in search efficiency. For example, studies show that the use of tabu lists with integer programming can save computing effort by at least two orders of magnitude for a given problem, as compared with standard integer programming Wang and Nie [2010]. To provide solution for complex optimisation problems, several hybrid algorithms have been developed by combining Tabu search with other metaheuristics Glover et al. [1995].

2.4.3 Memetic Algorithm

This algorithm was inspired by both Darwinian principles of natural evolution and Dawkins' notion of a meme, the term "Memetic Algorithm" (MA) was introduced in a technical report by Moscato et al. [1989]. MA are a class of stochastic global search heuristics in which Evolutionary Algorithms-based approaches are combined with local search techniques to improve the quality of solutions created by evolution Hart et al. [2005]. This serves as an extension of the global

Algorithm 2 Pseudocode for Memetic Algorithm

```
Encode solution space
Set population size,  $max_{gen}$ ,  $gen = 0$ 
Set crossover rate, mutation rate
Initialise population
while ( $gen < max_{gen}$ ) do
    Apply any generic global search metaheuristic approach.
    Apply any local search.
end while
Apply final local search to best chromosome.
```

metaheuristics that uses a local search strategy to prevents the likelihood of early convergence [Hart et al. \[2005\]](#). The pseudocode for Memetic Algorithms is shown in Algorithm 2.

In a memetic algorithm the population is initialised at random at each individual uses local search to improve its fitness. Individuals with higher fitness are passed to the next generation. Selection phase of MA is similar to that of standard genetic algorithm. Using crossover operators, the selected parent pair are mated together to obtain new individuals or offspring. The latter are enhanced using a local search technique. The role of local search in MA is to locate the local optimum more efficiently then the GA.

2.5 Clustering using Meta-Heuristic Algorithms

Clustering is a non-polynomial (NP hard problem that is ineffectively solved by traditional techniques. Clustering in WSNs is seen as a NP-hard problem due to dynamic nature of WSNs exhibits in the way clusters repetitively which can very difficult to model through mathematical methods. Traditional clustering algorithms suffer from non-uniformity in clusters and CH distribution [Hou et al. \[2010\]](#) and they are expensive algorithms. Recently, computational intelligence (CI) paradigms are used to cluster WSN. This subsection summarises the work done by CI paradigms to cluster WSNs.

2.5.1 Clustering Using Genetic Algorithm

In [Park et al. \[2012\]](#), authors proposed the use of GA to solve the optimising problem for selecting the best number of CHs. A 9-bit binary coded representation for chromosomes was proposed, where the bits contains value '1' which represents a CH and value '0' which represents an ordinary node. The fitness function was defined as given in Equation 2.18.

$$Fitness = w * (D_T - D_i) + (1 - w) * (N - H_i) \quad (2.18)$$

Where D_T is the total distance from all nodes to sink, D_i is the total distances from regular nodes to their cluster head, N is the total number of nodes, H_i is the number of cluster heads, and w is preset weight. The results showed that the cluster layout depends on the location of the BS (sink). More CHs are found to be elected when the BS is close to the centre of the network.

[Xu and Saadawi \[2001\]](#) used the same model as presented in [Park et al. \[2012\]](#), but with different mutation mechanism and sink location. It is shown that when the CH reached about 25% of the overall nodes, a better fitness value is obtained. However, the choice of CH was not based on remaining energy of nodes after each round. This could lead to network failure or disconnection in the sense that any node with low energy might be elected as a CH. After sometime, this can cause the CH to die gradually and so that part of the network is disconnected. In [Van Dam and Langendoen \[2003\]](#), authors improved the previous work of [Park et al. \[2012\]](#), and [Xu and Saadawi \[2001\]](#) by adding the residual energy to the fitness function calculation as shown in Equation 2.19.

$$Fitness = RE + SE + (w * (D_T - D_i)) + ((1 - w) * (N - H_i)) \quad (2.19)$$

RE is the total cluster heads' energy, and SE total energy needed to send data from cluster heads to sink. RE and SE are both added to Equation 2.18 in Equation 2.19 to avoid selecting CHs with low energy. The results were compared with LEACH showed proper distribution of clusters and significant improvement in the network lifetime [Van Dam and Langendoen \[2003\]](#). In [Dong et al. \[2005\]](#),

2. Energy Efficiency Mechanisms for WSNs

authors used GA to optimise the clustering problem based on minimising the energy consumption. In their model, the radio transmission technology was used in their calculations. The Radio Energy Dissipation system of equations is presented in Equation 2.20.

$$\begin{aligned}
 TE_{xy} &= E_e + \epsilon_s d_{xy}^2, d < d_0 \\
 TE_{xy} &= E_e + \epsilon_l d_{xy}^4, d > d_0 \\
 d_0 &= \sqrt{\frac{\epsilon_s}{\epsilon_l}}
 \end{aligned} \tag{2.20}$$

Where TE_{xy} is the total energy needed to transmit data from a sensor to its CH, d is the distance between the sensor and its CH, d_0 is the threshold distance. Given that Electronics Energy $E_e = 50$, energy due to free space model $\epsilon_s = 10pj/m_2$ and energy due to multipath model $\epsilon_l = 0.0013pj/m_4$, communication distance between sensor to its CH $d_{x,y}$ and can exist either as the free ($d_{x,y}^2$ power loss) or the multipath fading ($d_{x,y}^4$ power loss).

The fitness calculation depends on the distance between nodes, CHs and sinks. The GA's outcome was the optimal Cluster Heads. The base station then identifies the cluster members and the transmission schedule. Each CH is assumed to send directly to the sink. Although their algorithm performed better than LEACH, the improvement was not significant. This is because of the complexity of the fitness function. Lots of parameters were considered and each one is assigned a weight that is updated at each generation.

In Misra and Banerjee [2002], authors proposed a GA to minimise the communication distance. Moreover, a two-dimensional chromosome representation is used. The chromosome mapped the actual sensor layout of the deployment area. The gene's value of zero indicate non-existing nodes, '1' indicates a sensor node, and '2' indicate a CH. The algorithm used result of the LEACH as an initial condition to GA algorithm. The fitness function used is as follows:

$$Fitness = \sum_i \sum_j d_{CH(i,j)}^2 + \sum_j d_{SN(i)}^2 \tag{2.21}$$

where i is the number of CHs, j is the member number in cluster i , d_{CH} is

the distance between the sensor and its CH, and d_{SN} is the distance between CH and sink. The chromosome is divided into sectors, and crossover is performed by exchanging sectors between parents to ensure that the genes move with their neighbors. The results proved better performance than LEACH. However, the transmitted data size and cluster size is not added to the fitness function.

2.5.2 Clustering using Particle Swarm Optimisation

In Biswas and Morris [2005], authors applied particle swarm optimisation (PSO) to obtain the optimum cluster layout using a fitness function based on distance calculations (see Equation 2.22). Residual energy calculations were not included.

$$Fitness = \sum_{j=1}^k \sum_{i=1}^{n_j} n_j (d_{ij}^2 + \frac{D_j^2}{n_j}) \quad (2.22)$$

where d_{ij} is the distance between node i and its cluster head j , D_j is the distance from cluster head j to the base station, and n_j is the number of nodes in the cluster j . While varying inertia weight or the acceleration constant, the authors administered PSO algorithm. Analysis of the results are discussed in details in Biswas and Morris [2005]; Chachulski et al. [2007].

In Hou et al. [2010], authors proposed using an improved PSO algorithm to solve the uneven clustering problem. 5% of the total nodes were chosen to be the CH and each cluster has equal number of nodes. Their fitness was based on the communication distance. The PSO dynamic inertia weight was modified to include the particles' diversity. The CHs resulted from the PSO algorithm is then checked for their energy level. If their energy level falls below a threshold, they are replaced by the nearest node whose energy is more than the threshold. Compared with LEACH and improved LEACH, the proposed PSO algorithm showed better results. However, the overall nodes remaining energy and lifetime is not considered.

2.5.3 Clustering using Differential Evolution

Chakraborty et al. [2012] presented a differential evolution based routing algorithm for more than a thousand relay nodes such that the energy consumed by

the maximum energy-consuming relay node is minimised. However, the authors don't put into consideration the cluster formation. Inappropriate clustering may lead to serious energy inefficiency of the relay nodes.

A DE based clustering algorithm was presented in [Kuila and Jana \[2014\]](#) and this algorithm employs an efficient vector encoding scheme and an extra phase called local improvement to improve the performance of clustering operations. This paper derived a fitness function that takes care of energy consumption of both gateways and the sensor nodes. This DE-based algorithm is shown to converge faster than traditional DE and GA. However, the drawback is that it assumes the gateways can directly communicate with the base station which may not be realistic for a large area network.

2.6 Conclusion

Wireless sensors networks are composed of sensor nodes deployed randomly in a sensing field and each sensor have the capability to collect and route data to neighbouring sensor or sink. The vast applications of WSNs can be classified into three categories namely the environmental monitoring system (examples are indoor, agriculture, habitat, forest monitoring etc), human body monitoring and intelligent buildings. However, the design of WSNs are confronted with some requirements and challenges such as short transmission range, coverage problem, scalability issue, optimisation issue and limited energy. Several sleep scheduling and clustering techniques have been proposed by researchers to tackle this limited energy issue. The field of computation intelligence have also contributed to designing energy-efficient cluster-based WSNs by employing meta-heuristic approaches. This approach employs two search strategy for providing better solution for clustering problem, namely the global search (GA, PSO and DE), local search (SA, ILS, TS etc) and Memetic Search strategy. In the next chapter, a review of machine learning approaches that can applied to improve the accuracy of event-based WSNs applications such as fire detection system will be discussed.

Chapter 3

Intelligent Machine Learning Mechanisms for WSNs

This chapter discusses about the use of WSNs for monitoring or event detection applications such as fire detection system. Traditional fire detection techniques have the problem of false alarms and delay in fire detection. Recently, the introduction of wireless sensor network into fire detection systems has helped to improve the Quality-of-Service. The study of WSN-based fire detection system can be viewed from the operation of a single sensor node and the combination of several sensors in a distributed manner. Also, recent studies have tended towards incorporating artificial intelligence based techniques into wireless sensors for enhanced and intuitive performance. Current Artificial Intelligence algorithm uses two Machine learning approaches for its operation, which are namely the Supervised and Unsupervised learning approach. Finally, this chapter takes a critical study of the various AI-based techniques that can be used for the two approaches.

3.1 Introduction

Due to the incessant losses of lives and damage done to property due to fire mishap, people have come to realise that there is need for early detection of fire in its early stage. This foundational step will help in decreasing fire loss, en-

3. Intelligent Machine Learning Mechanisms for WSNs

sure safety of life and property by applying the fire extinguishing procedures. Therefore, researchers all over the globe have been trying out some new techniques that will help in fast and accurate fire detection. Fire is produced from a combustion process that has the potential of emanating into a huge disaster by randomly spreading out from its source especially when it is out of control. The design of an automatic fire system in order to detect fire in its premature stage works on the rules and principle of how fire is generated and the status information or alert sent to people for extinguishing actions. This is an ideal objective of fire detection. The traditional fire detection has the flaws of generating false alarms and the problem of reliability and adaptability.

Event detection is a newly discovered functionality of wireless sensor network apart from periodic monitoring, which involves the transfer of data without any observation of change. Event detection offers extended capability of when and where events of interest occur. The design of in-network event detection methods for wireless sensor networks is not an easy task, as there is a need to cope with various challenges and issues such as the unreliability, heterogeneity, adaptability and resource constraints. This survey paper tackles the problem of time critical event detection in wireless sensor networks by studying different fast, accurate and intelligent methods using artificial intelligence (AI). The intelligent wireless sensor network employs machine learning algorithms using the supervised and unsupervised approaches.

Wireless sensor networks (WSNs) are composed of a large number of small, cheap devices, called sensor nodes or motes, which are equipped with sensing, processing, and communication capabilities. Low-power micro-processors, sensor technology, and low-power RF design [Borbash \[2004\]](#) are the enabling technologies that converge to make the wireless sensor network. They have recently contributed to the development of various applications such as asset tracking, industrial automation, logistics, and health care [Lewis et al. \[2004\]](#). Applications of WSNs are synonymous to any typical organism in the way that they rely on sensor data being collected from the physical world [Lewis et al. \[2004\]](#). Data collection in WSNs can broadly be seen from three perspectives, namely [Chen and Varshney \[2004\]](#):

1. Continuous data collection: Data is collected by each sensor node and

3. Intelligent Machine Learning Mechanisms for WSNs

transmitted individually to a central point (base station) periodically. This method of data collection is simple but the only drawback is that WSNs does not work well when the data collection rate is very high. Data continuously sent to the base station is prone to low data resolution, network traffic, data latency, packet loss and high energy consumption.

2. Query-driven: Here, sensor nodes send query to the network for sensor data, and the request database is reply to the sensor nodes. This type of data collection is more efficient in terms of network traffic and energy consumption because sensor nodes only respond to explicit data request from neighbouring sensor Kasi et al. [2012]. However, applications that are driven by implicit, complex or unknown pattern event cannot benefit in the use of this type of data collection.
3. Event-driven: In this type of data collection, sensor nodes send data or trigger an alarm only when an abnormal situation occurs. This helps to saves considerable amount of data transmission and energy consumption compared with continuous and query-driven data collection. This is suitable for application where implicit and complex data that exhibit unknown patterns are sent across the WSNs.

In general, raw sensor data collected from sensors contain errors or missing values due to sensor failures, transmission errors, biases and other measurement errors. Therefore, there is need to pre-processed the corrupt or missing value raw data collected from the sensor node and turn it into useful information. Thereafter, appropriate decisions are made based on the useful information and finally the correct action applied. Generally speaking, data processing occurs in one of the following locations in WSNs:

- Base station: Here, raw data are gathered from different sensor nodes in a network. And it also serves as the interface between wireless sensor network and the outside world. The base station is usually equipped with stronger computational energy and more network resources compared with a sensor node. To accomplish the task of data processing, the base station has a higher memory, processing capability and energy sources but suffers data

3. Intelligent Machine Learning Mechanisms for WSNs

transmission related problems such as network traffic, packet loss and latency.

- In-network: Data pre-processing task is done by sensor node(s) rather than the base station. So, only the useful data is sent through the network. This helps to reduce the amount of traffic data in the network thereby solving problem such as network congestion, latency and packet loss. Generally speaking, in-network data processing can be conducted in two manners:
 - Local: only one sensor node in the network does the data processing task.
 - Distributed: Data processing task are done by a group of sensors. Consequently, sensor nodes should exchange some data.

The continuous data collection performs data processing task at the base station. Although, traditional query and event data application does data pre-processing at the base station, modern application is now tending towards in-network data processing. By performing in-network processing at the sensor nodes, decisions are made quickly without the need for instruction from the base station [Kasi et al. \[2012\]](#). Therefore, this survey paper is specifically interested in event-driven data collection that involves local and distributive in-network data processing. Event-driven data collection can synonymously be referred to as event detection. Event detection is one of the main components in numerous wireless sensor applications such as invasion of enemy forces, health monitoring, and fire detection [Tilak et al. \[2002\]](#). The remainder of this study is organised as follows. Section 3.2 examines intelligent WSNs based on machine learning approaches for event detection applications. Section 3.3 reviews the different supervised machine learning algorithms for various fire detection applications. Section 3.4 discusses two major unsupervised machine learning algorithms used in recognition of hidden patterns in event data. Finally, a conclusion is presented in Section 3.5.

3.2 Intelligent WSN-based Approach for Event Application

AI-based event detection techniques use artificial intelligence-based methods (also known as machine learning (ML)) to detect events. AI-based techniques are generally classified into two classes, this is the supervised and unsupervised. Supervised Machine learning or classification techniques use labelled data for training, while unsupervised learning or clustering techniques require no labelled data or a priori knowledge. AI-based methods are also pattern matching approaches as they look for data patterns or trend usually using a non-linear function. The most important advantage of AI-based techniques is that they do not require expert knowledge to set the parameters of approaches. Therefore, an AI-based technique attunes its parameter by learning automatically from data. These techniques are, in general, less computational intensive than model-based (especially statistical) approaches [Mitchell \[2006\]](#). Since events generally exhibit a pattern, learning-based techniques appear to be quite promising for accurate event detection in WSNs as they can learn data pattern without human intervention. The new trend in data driven decision making systems (e.g. event detection in WSNs) is, therefore the use of AI-based approaches [Mitchell \[1999\]](#). In choosing the right machine learning algorithm for specific application, there is need to compare and evaluate the following criteria:

- **Speed:** This refers to the computational costs involved in generating and using the given classifier, as they need to be implemented on tiny resource constraint sensor nodes.
- **Accuracy:** The accuracy of a classifier refers to the ability of a given classifier to correctly predict the outputs of new or test data.
- **Robustness:** This is the ability of classifier to make correct predictions in the presence of noisy data and missing values.
- **Scalability:** This refers to the ability to construct the classifier or predictor efficiently given large amounts of data.

- Interpretability: This refers to the level of understanding and insight that is provided by the classifier or predictor.

3.3 Supervised learning Algorithms

Supervised learning techniques strongly rely on presence of labeled data and require a training phase.

3.3.1 Artificial Neural Network

Artificial neural networks (ANN) are family of statistical learning algorithm that works in a similar fashion with biological neural networks (central nervous systems of animals in particular the brain), and are used to estimate functions that are generally unknown and dependent on large input values. An artificial neural network is an interconnected group of nodes called neurons, which computes input values that are capable of doing machine learning, pattern recognition or event detection. In Yu et al. [2005], a distributed neural network based event detection approach was presented, which demonstrated forest fire detection in WSNs. The in-network data processing was performed at both the sensor node and cluster head. The sensor use threshold-base event detection approach and the cluster head use neural network event detection for making reliable decision of the forest fire. This approach can be simulated for both small and large scale networks.

The Feed forward neural network (FFNN) is a type of ANN in which each layer is fed by its nearest layer. There are no feedback connections to previous layers Mesin et al. [2011]. FFNN is trained by using the back-propagation, an algorithm that learns by iteratively processing a training set and comparing the predicted output with the actual known target output. During the training process, the weights are modified in a backward direction so as to minimise the mean square error between the predicted and actual target value. This approach was use in Xue [2010] to identifying road tunnel fire by selecting the temperature, smoke density and the density of CO for road tunnel as BP neural network input variables. The

computation complexity of a FFNN is written as:

$$O_{FFNN} = O(m \times n \times p) \quad (3.1)$$

Where m neurons is the number of features, n is the number of neurons in the hidden layer, and p neurons in output layer.

3.3.2 Naive Bayes Classifier

A Naïve Bayes classifier is a simple probabilistic classifier that obtains the posterior probability of each class C_i using Bayes rule. The Naive Bayes classifier (NBC) makes the simplifying assumption that the attributes A , are independent given the class, so the likelihood can be obtained by the product of the individual conditional probabilities of each attribute given the class [Zhang \[2004\]](#). Thus, the posterior probability $P(C_i|A_1, \dots, A_n)$ is given by:

$$P(C_i|A_1, \dots, A_n) = \frac{P(C_i)P(A_1|C_i) \dots P(A_n|C_i)}{P(A)} \quad (3.2)$$

Where

- $P(C_i|A_1, \dots, A_n)$ is the probability of class C_i given the data attributes A_1, \dots, A_n , otherwise refers to as the posterior probability.
- $P(A_1|C_i) \dots P(A_n|C_i)$ is the probability of data attributes given that the class C_i was true.
- $P(C_i)$ is the probability of class C_i being true (regardless of the attribute A). This is called the prior probability of $P(C_i)$.
- $P(A)$ is the probability of the attribute A (regardless of the class C_i).

The main focus is to calculate the posterior probability of $P(C_i|A_1, \dots, A_n)$ from the prior probability $P(C_i)$ with $P(A)$ and $P(A_1|C_i) \dots P(A_n|C_i)$. After calculating the posterior probability for a number of different classes, you can select the class with the highest probability. Here, the probability table is made once and

3. Intelligent Machine Learning Mechanisms for WSNs

then programmed into the sensor nodes. The computational complexity is calculated for the search process only, which is more expensive. The computational cost is written as:

$$O_{NaiveBayes} = O(m \times i \times j) \quad (3.3)$$

Where calculated using m (number of features), i is number of classes and j is number of intervals. Taking into account the nature of the underlying probability model, the Naive Bayes classifier can be trained very efficiently in a supervised learning setting, working much better in many complex real-world situations, especially in the computer-aided diagnosis than one might expect [Gorunescu \[2011\]](#).

In theory, Bayesian classifiers have the minimum error rate when applied to a large database in comparison to all other classifiers. However, in practice this is not always the case, owing to inaccuracies in the assumptions made for its use, such as class-conditional independence, and the lack of available probability data. As mentioned before, two important problems for the NBC are how to deal with dependent and continuous attributes. In [Abidha and Mathai \[2013\]](#), NBC was introduced to an entropy-functional-based online adaptive decision fusion (EADF) framework as a new approach for computational vision-based fire and flame detection. The EADF was used to fuse a set of decisions made by sub-algorithm such as slow moving video object detection, Smoke-colored region detection, wavelet-transform-based region smoothness detection Shadow detection and elimination and covariance-matrix-based classification using NBC. The merit of Naive Bayes is that it requires a small number of training data to compute the means and variances that is used for classification. However, independent variables are assumed because only the variances for each labels is required and not the total covariance matrix [Akthar and Hahne \[2012\]](#).

3.3.3 Decision Tree

This event detection schemes construct a decision tree for data classification in its training phase by using a local search greedy algorithm. The tree should contain the minimum required nodes or depth to reduce time and memory complexities

Abidha and Mathai [2013]. The computational complexity of the algorithm is given as

$$O_{DecisionTree} = O(m \times i \times \log i) \quad (3.4)$$

Where m is the number of features and i is the number of training set. The result presented in Bahrepour et al. [2010], shows better detection accuracies for fire detection by decision tree classifier compared with neural network and Bayes classifiers. Decision tree performs well with large datasets and even if its assumptions are somewhat violated by the true model from which the data were generated. Also, decision tree can handle both numerical and categorical data.

Decision-tree learners can create over-complex trees that do not generalise well from the training data. There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems. Also for data stored in categorical variables with different number of levels, information gain in decision trees is biased in favour of those attributes with more levels.

3.4 Unsupervised Learning (Clustering) Algorithm

Unlike supervised learning techniques, unsupervised learning technique do not require labelled data, training phase, and a priori knowledge about event patterns.

3.4.1 Fixed-width clustering

In Eik Loo et al. [2006] authors propose a fixed-width clustering technique for intrusion detection. The idea of fixed width clustering is to cluster training data into a dynamic number of clusters in the training phase. Then, based on the population of clusters, a decision is made on which cluster belongs to the intrusions. After the training phase, the data which is closer to intrusion cluster is reported as intrusion attacks. Another fixed-width clustering technique is proposed in Rajasegarar et al. [2006]. The authors propose that single nodes cluster data

3. Intelligent Machine Learning Mechanisms for WSNs

and then in a sink (or cluster head) these clusters get merged together to detect anomalous data. The approach is implemented in C++ on Great Duck Island data (humidity, temperature and pressure sensor data) for detecting anomalous data.

The advantages of using technique are mentioned below and among them are:

1. It is easily adaptable to incremental mode (i.e. after learning the clusters, new points can be inserted into the system and tested for outliers).
2. It does not have to be supervised
3. It is suitable for anomaly detection from temporal data
4. The testing phase for clustering based techniques is fast since the number of clusters against which every test instance needs to be compared is a small constant [49].

The limitation of this technique is that in the presence of large data the result may be biased. Finding clusters of data objects in high dimensional space is challenging, especially considering that such data can be sparse and highly skewed. Most real-world databases contain outliers or missing, unknown, or erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

3.4.2 *k*-means Clustering

k-means clustering aims to divide a set of n observations ($x_j = x_1, x_2, \dots, x_n$) into $k(\leq)$ disjoint set so as to minimise the sum-of-squares criterion.

$$J = \sum_{j=1}^k \sum_{i=1}^n ||x_i^{(j)} - c_j||^2 \quad (3.5)$$

Where $||x_i^{(j)} - c_j||^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j , is an indicator of the distance of n data observations from their respective cluster centres. In general, the algorithm does not achieve a global minimum of J over the assignments. The limitation of this algorithm is that

the reachable minimum is not a proper local minimum because the algorithm uses discrete assignment rather than a set of continuous parameters. However, the algorithm is often used due to its ease of implementation [Shi et al. \[2010\]](#). According to [Jain \[2010\]](#), the main k -means clustering algorithm steps are as follows:

1. Select an initial partition with K clusters; repeat steps 2 and 3 until cluster membership stabilizes.
2. Generate a new partition by assigning each pattern to its closest cluster centre.
3. Compute new cluster centres.

3.5 Conclusion

This chapter highlights the use of artificial intelligence with wireless sensor network for prompt and effective fire detection. Different intelligent WSN-based techniques were examined based on the supervised and the unsupervised learning algorithm. However, unsupervised learning approaches are still immature for event detection in WSNs. The clustering technique is a challenging field of research that has lots of potential for application that requires accurate and fast detection of unknown events accurately and can also give semantic to event autonomously. There are some parameters in supervised learning approaches (e.g., weights in artificial neural networks) that can be adaptive when (i.) the same type of events need to be detected in another environment, or (ii) a set of different events need to be detected in the same or another environment. Designing an expert system which can make supervised event detection adaptive to aforementioned situations is another future direction of this study. In the next chapter, a hybrid approach that combine k -means with neural networks and other classifiers in order to improve the detection rate of event detection applications is proposed.

Chapter 4

Proposed Machine Learning Approaches for WSNs

This chapter presents a new hybrid approach that involves the use of k -means algorithm with neural networks, an efficient supervised learning algorithm that extracts patterns and detects trends that are hidden in complex data. Previous research on event detection concentrates majorly on the use of feed forward neural network and other classifiers such as naive Bayes and decision tree alone for modern fire detection applications. In this approach presented here, k -means combines with neural networks and other classifiers in order to improve the detection rate of event detection applications. To demonstrate this approach, data aggregation was performed on normalized multi-dimensional fire datasets in order to remove redundant data. The aggregated data forms two clusters which represent the two class labels (actual outputs) with the aid of k -means clustering. The resulting data outputs are trained by the Feed Forward Neural Network, Naive Bayes, and Decision Trees. This approach was found to significantly improve fire detection performance.

4.1 Introduction

There is a growing need for prompt and accurate fire detection in any extensive indoor environment, and this can only be achieved by employing appropriate

4. Proposed Machine Learning Approaches for WSNs

wireless sensor network (WSN) deployment. WSN ad-hoc networks are made of tiny wireless nodes called sensor nodes that measure any event or exceptional change in environmental conditions. The sensors organise themselves into a single multi-hop or hierarchical network structures with several clusters and cluster heads. Each sensor node is capable of sensing, processing, and transmitting data to the base station [Shen et al. \[2001\]](#). In composite event detection systems such as fire alarms, the two foremost goals are speed and accuracy. One way to achieve these goals is by performing data aggregation at central nodes. This helps reduce energy consumption and redundancy.

A composite event is the combination of different observation of attributes. For an event fire alarm application to make a decision of normal or abnormal situation, there may be the need to combine several attributes based on large number of sensor types (temperature, carbon monoxide (CO), smoke) which are spatially distributed over a wide area [Memon and Muntean \[2012\]](#). Data obtained from a composite event are multidimensional in nature. One of the key measures of enhancing accurate fire detection decisions is to perform data aggregation at intermediate nodes or at the clustered head. Data aggregation usually involves the fusion of data from multiple sensors at intermediate nodes and transmission of the aggregated data to the base station (sink). Data aggregation helps to remove redundant and highly correlated data generated from neighboring sensors at the intermediate node before transmission to the base station [Rajagopalan and Varshney \[2006\]](#). Data aggregation techniques are also very effective in reducing communication overhead by collecting the most critical data from the sensors and making it available to the sink in an energy efficient manner with minimum data latency. Data latency is a crucial requirement in most event detection application such as fire detection applications.

The aggregated data at the clustered head are sent to the decision center (base station) for a final decision using an appropriate machine learning algorithm e.g. to detect the event status. This machine learning algorithm is divided into supervised and unsupervised learning approaches. The former approach relies on the presence of labeled data and a training phase whereas the latter approach does not require labeled data, training and prior knowledge of the event patterns [Kotsiantis \[2007\]](#); [Nagpal et al. \[2013\]](#). The accuracy of supervised learning is

4. Proposed Machine Learning Approaches for WSNs

often negatively affected when many attributes depend on one another. Learning large and complex models has increased difficulty with supervised learning than with unsupervised learning.

This chapter presents a new hybrid approach that involves the use of k -means algorithm with neural networks, an efficient supervised learning algorithm that extracts patterns and detects trends that are hidden in complex data. Previous research on event detection concentrates majorly on the use of feed forward neural network and other classifiers such as Naive Bayes Classifier (NBC) and Decision Tree (DT) alone for modern fire detection applications. The remainder of this study is organised as follows. Section 4.2 briefly reviews contributions for fire detection mechanism using WSN. Section 4.3 discusses data aggregation techniques in cluster-based WSNs. Section 4.4 proposes a new hybrid learning approach that combines k -means with FFNN, NB and DT classifiers. Section 4.5 presents empirical results that depict the performance of our system approach using some test data. Finally, a conclusion is presented in Section 4.6.

4.2 Background

According to the National Fire Danger Rating System (NFDRS) for forest fire detection, four sensor types (temperature, humidity, smoke and wind speed) were used to generate a fire-likelihood index [Yu et al. \[2005\]](#). The contribution of this study is the function of a feed-forward neural network (FFNN) and data aggregation for reducing overhead communication. A system approach was proposed in [Zhiping et al. \[2006\]](#) for forest fire detection using sensor nodes, gateway(s) and task manager(s). The sensor types used were temperature and humidity. Data obtained from different sensor nodes were fused together at the gateways. The data analysis and decisions are taken at the task manager.

Early fire detection in open spaces such as forests and urban areas was proposed using a sensor network approach [Zervas et al. \[2007\]](#). For early fire detection, the authors used temperature sensor and maximum likelihood (ML) to fuse sensor data. Their system architecture is made up of the sensing, computing and localised alerting unit. According to previous works on fire detection using WSN, it is found that accurate and early fire detection can be approached from two per-

4. Proposed Machine Learning Approaches for WSNs

spectives [Cestari et al. \[2005\]](#). Firstly, sensor data from several nodes of the same sensor type can be aggregated into one. Secondly, an artificial intelligence (AI) can be incorporated to recognise patterns in that data. Generally, the selection of sensors is based on random process or assumption. Researcher have discovered that the use of a single sensor type such as temperature sensor cannot guarantee accurate fire detection and so, there is need to employ multi-sensor type fire detectors that are capable of monitoring the environment against any changes in the amount of carbon monoxide (CO), carbon dioxide (CO₂) and oxidised gas. The use of multi-sensor will help to provide more accurate fire detection decision and discrimination between fire and noise [Cestari et al. \[2005\]](#); [James \[1999\]](#).

In the proposed hybrid approach, four optimal sensor types were selected, which are temperature, ionization, photoelectric and CO sensors. The flaming fire and smoldering fire are detected by the ionization and photoelectric sensors respectively. A two-storey building example is used. It is assumed that every node in the WSN contains all the required sensors. At intermediate stages, data aggregation was performed on the continuous data obtained from different sensor points of the same type. Data aggregation helped to avoid communication overhead between neighboring nodes. *k*-means clustering was subsequently performed to divide the aggregated data of the selected optimal sensor into two clusters. And finally, efficient and cheap detection algorithms such as the feed forward neural network (FFNN), Naïve Bayes and Decision Tree are employed to improve the performance accuracy significantly.

4.3 Data Aggregation in Clustered-Based WSNs

In some WSNs where sensor nodes are densely deployed, each sensor node senses similar data from the physical environment due to closeness of sensor nodes. This type of sensor network will result in transmitting redundant data and this has the potentials to degrade the overall network. To solve this issue, there is a need to perform some grouping of sensor nodes and also combining or compressing data and transmitting only the compact data [Maraiya et al. \[2011\]](#).

In cluster-based WSN, sensors are grouped in clusters and in-network data aggregation is done locally within the clusters. A cluster head plays the role

4. Proposed Machine Learning Approaches for WSNs

of aggregator which aggregate data received from cluster members locally and then transmits the result to base station (sink). In heterogeneous networks, the clustered head or aggregator node has a higher energy capability compare with the member sensor nodes within the same cluster network. A data aggregation scheme is energy efficient if it maximises the functionality of the WSN in the sense that sensor nodes should spend the same amount of energy in every data gathering round. Figure 1 below shows a cluster-based sensor network organisation that involves a long range transmissions or multi hopping through other cluster heads to the sink or base station.

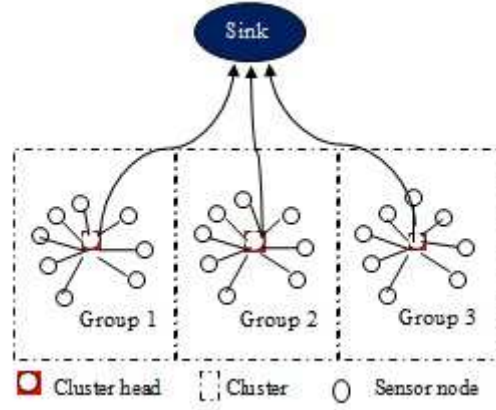


Figure 4.1: Cluster based sensor network

The performance measures of data aggregation in cluster base WSN for event detection application are discussed below:

- **Network lifetime:** Network lifetime is defined as the number of data aggregation rounds till certain percentage of the sensors die. In applications such as fire detection where all sensor nodes are vital, the lifetime is define as the number of rounds until the first sensor drains off its battery energy or dies completely
- **Data Accuracy:** This is the evaluation of the ratio of the total number of reading received at the base station to the actual total number of data generated.

- Latency: Latency is defined as the delay involved in data transmission, routing and data aggregation. It can be measured as the time delay between the data packets received at the sink and the data generated at the source nodes.

4.4 Proposed Hybrid Learning Approach

In this hybrid approach, the k -means clustering algorithm is used to assign labels to a set of features from four sensor types (temperature, ionization, photoelectric and CO). k -means clustering was used to partitioned the data into two clusters (i.e. $K=2$), and assigned labels to the data. The labeled data output is fed into the classifier for training and testing purposes. The FFNN, Naïve Bayes and Decision tree are used as classifier and the prediction accuracy was presented in section 4.5. In this work, k -means clustering (refer to section 3.4.2 of Chapter 3) will be combined with FFNN, Naïve Bayes and Decision Tree classifier for accurate fire event detection. The major work of the classifier is to accurately predict the class (fire or non-fire) of each data instance based on the combination of processed data obtained from the four sensor types.

Feed Forward Neural Network

Feed forward neural network (FFNN) is a type of the neural network, in which each layer is fed by its back-layer [Mehrotra et al. \[1997\]](#). FFNN consists of one

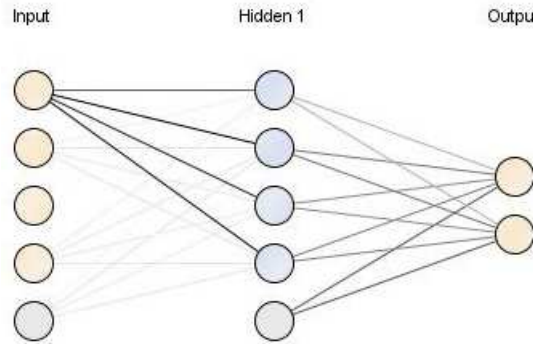


Figure 4.2: Diagram of Feed Forward Neural Network

4. Proposed Machine Learning Approaches for WSNs

input layer, one or more hidden layers and one output layer. Figure 4.2 shows the FFNN's architecture. FFNN is trained by learning iteratively processing a training set and comparing the predicted output with the known target output using back-propagation algorithm. During the training process, the mean square error between the predicted and actual target value is minimised by adjusting the weights in a backward direction. One major challenge of FFNN is finding the optimal weight and one of the ways of finding the weights is through gradient descent (GD) approach Bahrepour et al. [2009].

Naive Bayes Classifier

In this proposed approach, the NBC model assumes that the presence (absence) of a particular feature does not relate to the presence (absence) of any other feature. According to equation 3.2, the posterior probability of class (fire and non-fire or noise, which are represented in Figure 4.3 as Cluster_0 and Cluster_1 respectively) given by a feature is the product of probability of feature given that the class is true and the probability of a class being true regardless of the features divided by the probability of the feature regardless of the class. Considering each feature, the posterior probability for each class is computed using Equation 3.2; for example the posterior probability for each class considering temperature only is computed as:

$$P(Fire|Temperature) = \frac{P(Fire)P(Temperature|Fire)}{P(Temperature)} \quad (4.1)$$

$$P(Non - fire|Temperature) = \frac{P(Non - fire)P(Temperature|Non - fire)}{P(Temperature)} \quad (4.2)$$

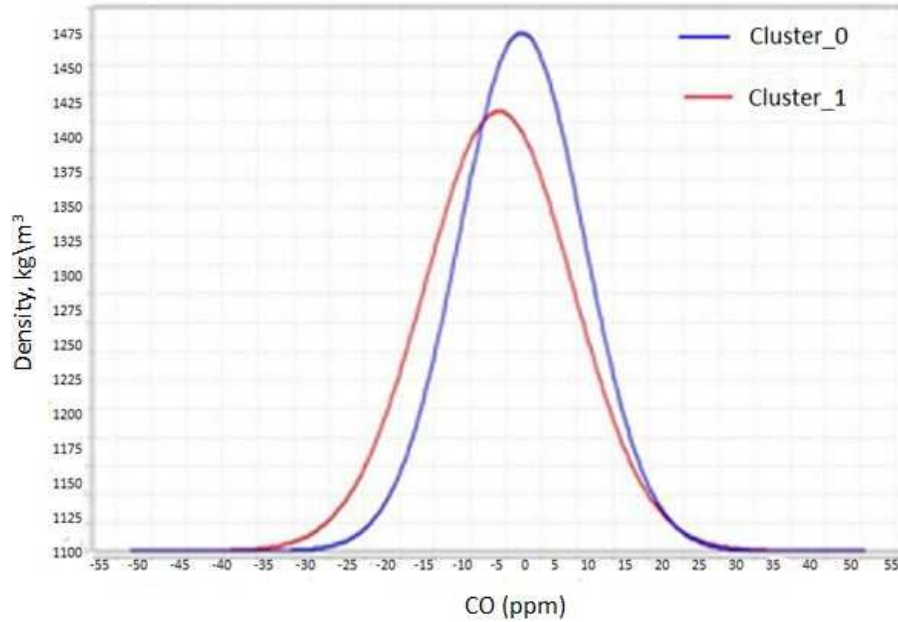
Considering all features, the posterior probabilities of each class are multiplied together as shown below:

$$P(Fire|Temperature, CO, Ion, Photo) = P(Fire|Temperature) \times P(Fire|CO) \times P(Fire|Ion) \times P(Fire|Photo) \quad (4.3)$$

4. Proposed Machine Learning Approaches for WSNs

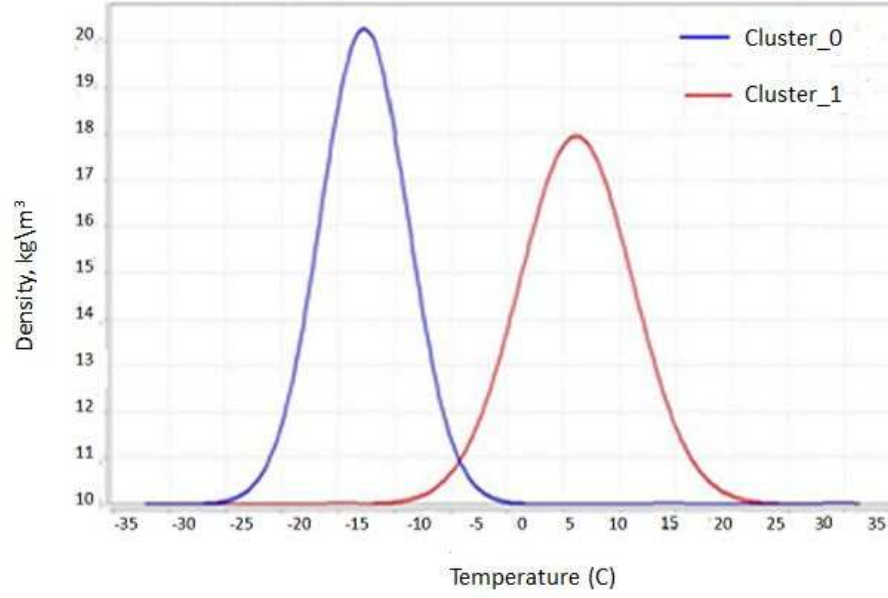
$$P(Non-fire|Temperature, CO, Ion, Photo) = P(Non-fire|Temperature) \times P(Non-fire|CO) \times P(Non-fire|Ion) \times P(Non-fire|Photo) \quad (4.4)$$

After calculating the posterior probability for each of the two classes (fire and non-fire) as shown in Equation 4.3 and 4.4, the one with highest probability is selected. In the empirical results described in section 4.5, the naive model is applied for sense data obtained from four sensor nodes or features (CO, temperature, photoelectric and ionization) are assumed to be independent of each other (i.e. they don't give any information about each other) and typically the distributions are assumed to be fixed. Based on these assumptions, the training portion for the NBC consists of the means and standard deviations for each of the input separately as shown in Table 4.1.

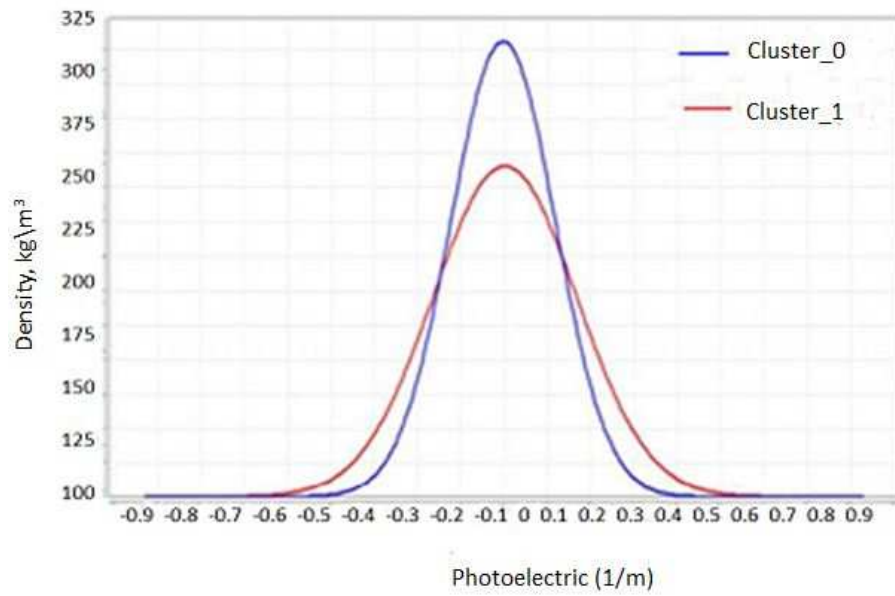


(a) Density Curves of CO

4. Proposed Machine Learning Approaches for WSNs

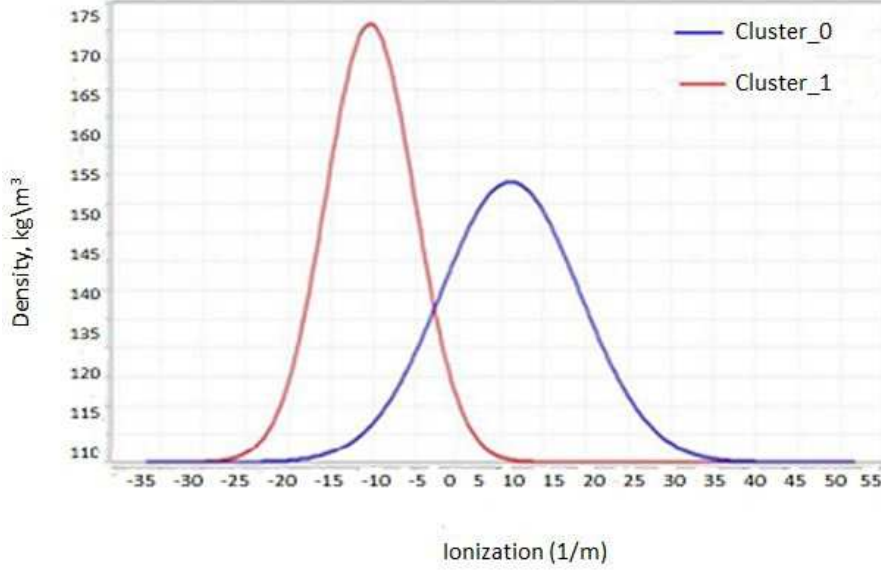


(b) Density Curves of Temperature



(c) Density Curves of Photoelectric

4. Proposed Machine Learning Approaches for WSNs



(d) Density Curves of Ionization

Figure 4.3: Density Curves of CO, Temperature, Photoelectric and Ionization respectively.

Also, the symmetrical density curves for CO, Temperature, photoelectric and ionization sensors are displayed in Figure 4.3, where the red and blue density curves denotes a fire and non-fire event respectively. For the density curve of temperature sensor data, it can be observed that the fire (cluster _0) area of the overlapped section with the noise (cluster _1) curve is smaller than the other sensors, whereas the area of overlapped section for the photoelectric sensor is the largest. This shows that the temperature is the most contributing attribute for fire prediction while the photoelectric sensor is the least contributor due to

Table 4.1: Distribution table for the four sensor types

Attributes	Cluster 0		Cluster 1	
	Mean	Standard Deviation	Mean	Standard Deviation
Temperature	-1.271	0.388	0.614	0.502
CO	0.216	0.880	-0.104	1.038
Ion	1.113	0.817	-0.538	0.523
Photo	-0.003	0.150	0.001	0.166

4. Proposed Machine Learning Approaches for WSNs

the obvious overlap between the fire and noise density curves. In Table 4.1, the photoelectric data attributes has the same approximate standard deviation of 0.2 for cluster 0 and cluster 1.

Decision tree

A decision tree is an inverted tree-like model because of its top root and bottom branches structure. The goal of this model is to predict the value of a target attribute called class or labels based on several input attributes of the datasets. In Rapid Miner TM an attribute with label role is predicted by the Decision Tree operator.

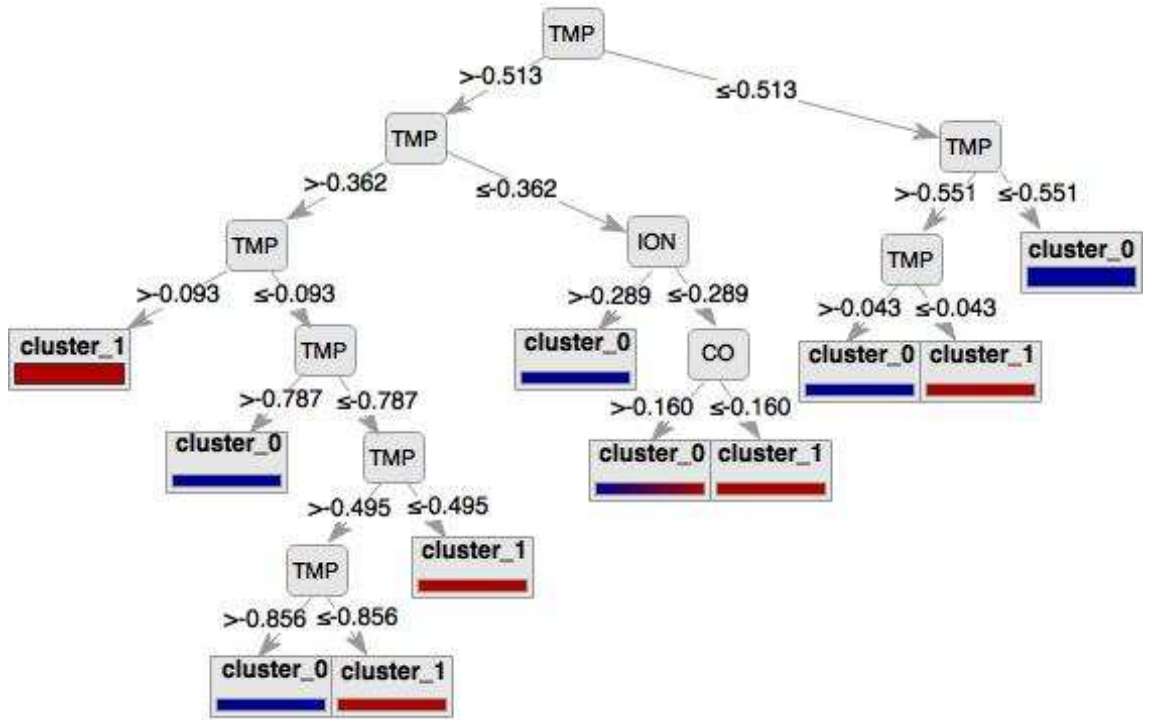


Figure 4.4: Diagram of Decision Tree

Note: In figure 4.4 above, the temperature, carbon monoxide and ionization sensors are represented as TMP, CO, ION respectively.

Each interior node of tree is matched to input attributes. The number of possible values of the input attribute is equal to the number of edges of nominal

interior node. Disjoint ranges label is assigned to outgoing edges of numerical attributes. Each leaf node describes the value of label attribute given the values of the input attributes represented by the path from the root to the leaf. The advantage of decision tree is that data representation is meaningful and easy to interpret compared with other approaches [Akthar and Hahne \[2012\]](#).

4.5 Empirical Results

To evaluate the prediction accuracy of the hybrid combination of k -means clustering with FFNN, Naive Bayes and Decision trees against the use of the classifiers alone, a set of data were obtained and a number of experiments were conducted. To evaluate the performance of the hybrid approach, experiments are carried out on six fire datasets obtained from NIST website (<http://smokealarm.nist.gov/>) namely two soldering fire dataset, two flaming fire dataset and two nuisance resource dataset. This dataset are merged together and pre-processed into a total of 1400 data instances with hundreds of attributes (based on sensor types and range), all having same units. All data from the same sensor type were fused together using the average operator. The aggregated data was grouped into two clusters and finally passed to the classifier.

The goal of the classifier is to accurately separate the data and classify them into their respective class, i.e., fire and non-fire (noise). The aggregated data obtained from the four sensor nodes (CO, temperature, photoelectric and ionization). After labelling, the data were passed to the classifiers. To perform a cross validation, 1400 data instances were divided to a 1000 training data and a 400 test data in order to ensure a fair comparison with the results presented in [Bahrepour et al. \[2009\]](#). All data were mixed at random and feed into the

Table 4.2: Empirical Results for all Classifiers

Hybrid Approach	Predictive Accuracy (Percentage)	
	Fire (Cluster 0)	Noise (Cluster 1)
FFNN	100.00%	99.79%
Naive Bayes	100.00%	99.26%
Decision Tree	98.26%	99.68%

classifier. Each test was repeated ten times and the mean prediction accuracy was recorded for each classifier type as shown in Table 4.2. Table 4.2 shows the prediction accuracy of the classifiers can be improved by using this hybrid approach. The accuracy of FFNN and Naive Bayes classifier in predicting the fire class is 100%, which is a significant improvement compared to 97.49% prediction accuracy obtained in Bahrepour et al. [2009]. Although, the prediction accuracy of the decision tree classifier is slightly lower than the other classifiers, nevertheless the prediction accuracy is still very good. The result shows that the three classifiers can accurately predict fire and non-fire situations.

4.6 Conclusion

This chapter have proposed and demonstrated the use of data aggregation to combine multidimensional data obtained from sensor nodes to reduce the data complexity and communication overhead. The hybrid combination of k -means clustering and the three popular classification approaches such as the FFNN, Naive Bayes and decision tree has enable us to generate a better fire prediction accuracy against the use of only the classifiers. The next chapter proposes a dynamic local-based algorithm that employs iterated local search operator with perturbation operator to obtain a better solution to solve energy optimisation problem in WSNs.

Chapter 5

Proposed Dynamic Local Search-Based Algorithm

In this chapter, a new clustering protocol employing an iterated local search (ILS) to solve cluster head selection problem is proposed. ILS uses a perturbation operator to change an initial random solution to produce a new point in the vicinity of the solution. Using a combination operator, this new point is mated with the random solution producing a new solution. A move from the current solution to the new solution is considered acceptable only for higher fitness value. If a move is rejected after a predetermined search length, the change rate of the current solution is increased in order to explore a wider search space for quality solutions. In each round, this search process continues until good solution that ensures balanced energy consumption is obtained for the network. Furthermore, a sleep scheduling scheme inspired by the Boltzmann Selection process in genetic algorithms is proposed. This mechanism stochastically considers coverage effect in the selection of nodes that are required to go into sleep mode in order to conserve energy of sensor nodes. The proposed mechanism of inactive node and cluster head selection protocols are performed sequentially at every round and they form part of the main algorithm proposed, namely the Dynamic Local Search-Based Algorithm for Clustering Hierarchy (DLSACH). The ultimate goal of the DLSACH protocol is to extend the network lifetime of wireless sensor networks by reducing and balancing the energy consumption among sensor nodes during communica-

tion processes. The proposed DLSACH protocol shows an improved performance compared to state-of-the-art protocols such as LEACH, TCAC and SEECH in terms of improved network lifetime for wireless sensor networks deployment.

5.1 Introduction

Recent progress in wireless communications and micro-electronics have contributed to the development of sensor nodes that are agile, autonomous, self-aware and self-configurable. These sensor nodes are densely deployed through-out a spatial region in order to sense particular event or abnormal environmental conditions such as moisture, motion, heat, smoke, pressure etc in the form of data [Oladimeji et al. \[2016\]](#). These sensors, when in large numbers, can be networked and deployed in remote and hostile environments enabling sustained wireless sensor network (WSN) connectivity. Hitherto WSNs have been used in many military and civil applications, for example, in target field imaging, event detection, weather monitoring, tactile and security observation scenarios [Naeimi et al. \[2012\]](#). Nevertheless, sensor node distribution and network longevity are constrained by energy supply and bandwidth requirements. These noted constraints mixed with the common deployment of large numbers of sensor nodes must be considered when a WSN network topology is to be deployed. The design of energy efficient scheme is a major challenge especially in the domain of routing, which is one of the key functions of the WSNs [Chakraborty et al. \[2011\]](#). Therefore, inventive techniques which reduce or eliminate energy inadequacies that would normally shorten the lifetime of the network are necessary. This chapter present a method which balances energy consumption among sensor nodes to prolong WSN lifetime. Energy resourcefulness is uniquely obtained using two described mechanisms; firstly, cluster head (CH) selection using a generic algorithm (GA) is employed that ensures appropriately distributed nodes with higher energies will be selected as CHs. Secondly, a Boltzmann inspired selection mechanism was utilised to select nodes to send into sleep mode without causing an adverse effect on the coverage.

The commonest routing protocols deployed to address the challenges discussed above are generally categorised into two classes, namely flat and hierarchical. Flat protocols comprise the well-known Direct Transmission (DT) and Minimum

5. Proposed Dynamic Local Search-Based Algorithm

Transmission Energy (MTE), which do not provide balanced sensor energy distributions in a WSN. The disadvantage of the MTE is that a remote sensor normally employs a relay sensor when transmitting data to/from the sink and this results in the relay sensor being the first node to die. In the DT protocol, the sink communicates directly with sensors and this results in the death of the remote sensor first. Consequently when creating WSNs, energy-efficient clustering protocols act as a pivotal factor for sensor lifetime extension. Generally, clustering protocols can perform better than flat protocols in terms of balancing energy consumption and network lifetime prolongation by employing data aggregation mechanisms [Abbasi and Younis \[2007\]](#); [Heinzelman et al. \[2002\]](#). In WSNs, there are three types of nodes considered: the cluster-head (CH), member node (MN) and sink node (SN). The member node manages sensing of the raw data and utilises Time Domain Multiple Access (TDMA) scheduling to send the raw data to the CH. The CH must aggregate data received from MNs and forward the aggregated data to the SN through single-hop or multi-hop. CH selection can be carried out by the sensors individually, by the SN or can be pre-implemented by the wireless network designer. Here, CH selection is performed by the SN due to the fact that the SN has sufficient energy and can perform multifaceted calculations.

In this thesis, the CH selection issue is viewed as an optimisation issue where the methods have employed a meta-heuristic local search strategy that involve the use of iterated local search operator combined with perturbation and combination operators to solve. These operators are applied on a set of random solutions in order to obtain the best solution within the search space. The quality of the solutions are accessed using a defined objective function. In this chapter, a Dynamic Local Search-Based Algorithm for Clustering Hierarchy (DLSACH) protocol was developed for clustering WSNs. This protocol performs two major operations; (1.) the use of an iterated local search algorithm for cluster head selection (ILSACHS) protocol to select the best cluster configuration and (2.) the proposed Stochastic Selection of Inactive Nodes (SSIN) mechanism in order to send some nodes that have negligible effect on the WSNs coverage to sleep or inactive mode.

In the proposed ILSACHS protocol, the sink performs the CH selection task on active nodes using the iterated local search with perturbation operator. The action of perturbation operator resembles the traditional mutation operator, and

5. Proposed Dynamic Local Search-Based Algorithm

is applied to a random solution at a specific change rate in order to generate a perturbed point in the neighbouring area of the random solution [Oladimeji et al. \[2016\]](#). The perturbed point is mated with a random solution and a new solution is produced using the combination operator. The fitness of the new solution is evaluated by the objective function. A move to the new solution from the current solution is accepted if the fitness value of the new solution is greater than the current solution, otherwise the move is rejected. The total number of search lengths for a good solution is divided into four step sizes. In the case where a move to a new solution is rejected after the last attempt of each step size division, the change rate is increased to explore wider search space.

At each network operation round, the search continues until a move to a new solution is accepted. This new solution is applied to the WSNs and is expected to contain distributed CHs that balances energy consumption across the networks. The SSIN, a mechanism that mimics the Boltzmann selection process in genetic algorithm (GA) was employed to reduce the number of active nodes at the beginning of each network operation round by sending some nodes to sleep or into inactive mode to conserve energy and prolong network lifetime with minor effects on coverage. Both mechanism works collaboratively to maximise network lifetime by balancing the energy consumption among sensor nodes during communication processes. The balance in energy consumption is achieved by selecting spatially distributed nodes with higher energy as CHs and also sending some nodes to sleep mode without causing an adverse effect on the coverage. The proposed DLSACH protocol is a more energy efficient protocol compared with other protocols.

The remainder of this study is organised as follows. Section 5.2 describes the network and energy dissipation model underlying the proposed protocol. Section 5.3 describes the objective function for the proposed protocol. Section 5.4 describes the proposed sleep scheduling mechanism, clustering algorithm and energy consumption computation. Section 5.6 discusses the experimental settings, performance measures, result and discussion. Finally, a conclusion is presented in Section 5.7.

5.2 Network and Radio Model

In this work, important network and radio model assumptions are adopted for the proposed algorithms and presented as follows:

- The data sink is a stationary and resource-rich device that is placed far away from the sensing field.
- All sensors are stationary after deployment and average energy is constant in either homogeneous or heterogeneous environment.
- All sensors have GPS or other location determination devices attached to them.
- Nodes are able to perform in inactive mode or a low power sleeping mode.
- Nodes that are close to each other have correlated data.
- The communication channel considered is assumed symmetric (i.e. the energy needed to transmit data from sensor node s1 to sensor node s2 is equal to the energy required to transmit a message from node s2 to node s1 for a particular signal to noise ratio (SNR)).

To ensure just comparison with previous protocols [Heinzelman et al. \[2002\]](#); [Liu et al. \[2011\]](#); [Vijayvargiya and Shrivastava \[2012\]](#), this thesis employed a simple radio energy dissipation model whereby the transmitter loses energy $E_{Tx}(k, d)$ to manage the radio electronics and the power amplifier, and the receiver dissipates energy $E_{Rx}(k)$ when managing the radio electronics, as shown in Figure [5.1](#). To transmit a k -bit message a distance d , the radio spends:

$$E_{Tx}(k, d) = \begin{cases} kE_{elect} + \varepsilon_{mp}kd^4, & \text{if } d > d_0 \\ kE_{elect} + \varepsilon_{fs}kd^2, & \text{if } d < d_0 \end{cases} \quad (5.1)$$

And to receive k -bit message, the radio uses:

$$E_{Rx}(k) = kE_{elect} \quad (5.2)$$

5. Proposed Dynamic Local Search-Based Algorithm

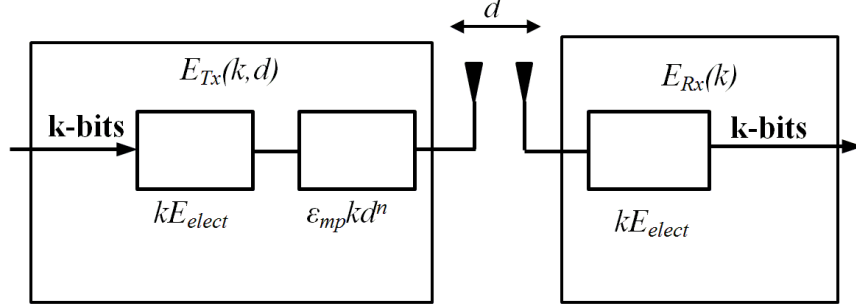


Figure 5.1: Radio Energy Dissipation Model

Where ϵ_{fs} is the free space model of transmitter amplifier, ϵ_{mp} is the multipath model of the transmitter amplifier, threshold distance $d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$ and the electronics energy E_{elect} depends on factors such as the digital coding, modulation, filtering, and spreading of the signal effect. Depending on the communication distance (d) between the transmitter and receiver, either the free space (d^2 power loss) or the multipath fading (d^4 power loss) channel models were used for all experiments. To use the free space (fs) model, the power-amplifier is fine-tuned such that the communication distance (d) is less than a threshold distance (d_0); else, the multipath (mp) model is used. ϵ_{mp} or ϵ_{fs} depends on the distance to the receiver and the acceptable bit-error rate.

5.3 Proposed Objective function

To solve the CH selection problem, objective functions are developed because CH selection is considered an optimisation problem. These objective functions return fitness values which are employed to assess the quality of a candidate solution. An objective function is found by taking into account parameters such as the total sensor node energy and the Risk penalty R . The sensor node energy parameter is considered to ensure that nodes with greater energy are given higher priority in the CH selection process.

5. Proposed Dynamic Local Search-Based Algorithm

The Risk penalty, R for the CH selection is defined as:

$$R = \begin{cases} Lower - L, & \text{if } L < Lower \\ L - Upper, & \text{if } L > Upper \\ 0, & \text{otherwise} \end{cases} \quad (5.3)$$

Based on several iterative tests, the percentage of CHs number (L) to the total number of sensor nodes (n) in the field proves to always give an optimal result between the *Lower* limit of 4% and *Upper* limit of 6%. Restrictions are imposed on the number of CHs using the parameter R .

Subsequently, the objective function is computed using:

$$F(X) = w_1 * \frac{AvgENCH}{AvgECH} + w_2 * R \quad (5.4)$$

Where w_1 and w_2 are the weighting factors. The average energy of non-CHs, $AvgENCH$ is the energy summation of all member nodes divided by the total number of member nodes ($n - L$) as given below:

$$AvgENCH = \frac{\sum_{i \in NCH} E_i}{n - L} \quad (5.5)$$

Also, the average energy of CHs, $AvgECH$ is the energy summation of all CH nodes divided by the total number of CHs (L) as given below:

$$AvgECH = \frac{\sum_{i \in CH} E_i}{L} \quad (5.6)$$

In equation 5.4, the ratio $\frac{AvgENCH}{AvgECH}$ is given a higher weighting factor ($w_1=0.9$) than the Risk penalty, R ($w_2=0.1$) because of its importance. (Note: CH and NCH represent the set of all CHs and non-CHs respectively).

5.4 The Proposed DLSACH protocol

In this protocol, the sleep scheduling, clustering and energy consumption computations are performed in succession. We propose the stochastic selection of inactive node (**SSIN**), a sleep scheduling scheme is used to put some sensor nodes

5. Proposed Dynamic Local Search-Based Algorithm

Algorithm 3 DLSACH Protocol

Let *AliveNodes* be the total number of sensor nodes
Compute the network total coverage.
while (*AliveNodes* > 0) **do**
 Use algorithm SSIN to select inactive nodes. (See Algorithm 4)
 Put selected nodes into sleep mode.
 Apply the proposed iterated local search algorithm for CHs selection. (See Algorithm 5)
 Compute the energy values of E_{CH} , E_{Mem} and E_{Res} . (refer to Section 2.1.3)
 Find out the number of dead nodes (node with energy equal or less than 0).
 Update *AliveNodes*
end while

into sleep mode without harming the functionality of WSNs in terms of network coverage. The sleeping scheduling is performed at the beginning of every round during network operations. In the clustering process, the proposed ILSACHS protocol works in such a way that a local solution with a known fitness value is obtained from a set of random solutions. The perturbation operator mutates the selected local solution at a specified mutation rate in order to obtain a point. Using the combination operator, this point is mated with the local solution to produce a new local solution. The fitness value of the new local solution is obtained by evaluating the objective functions and compared with the previous solution. If the fitness value of the new solution is greater than the previous one, a move to the new local solution is accepted otherwise the new solution is discarded. The cycle continues until the moves reach the local optimal solution within the specified search length. In the case where no moves are accepted after a certain search length, the mutation rate of the perturbation operator is increased according to defined step sizes to widen the search length. At each network operation round, the final solution obtained in this protocol is expected to minimise the energy consumption due to well distributed CHs in the network. The energy consumption computation is performed at each network operation round as it moves from the set-up to steady state phase. At the setup phase, the sink transmits control packets to receive node information in terms of the

5. Proposed Dynamic Local Search-Based Algorithm

nodes ID, location and energy. The residual energy of each sensor is computed at the end of each round of the steady state of network operation. The value of the residual energy computed in the current round is used for the next round as a parameter for sleep scheduling and cluster head selection process. This cycles continues until all nodes are dead; as shown in Algorithm 3.

5.4.1 Proposed SSIN Mechanisms

In this section, the estimation of coverage by setting up a matrix that computes the number of nodes covering the area within each grid point is discussed. Furthermore, SSIN protocol that uses the energy values and coverage effect in deciding which nodes to send into sleep mode is presented.

Coverage Estimation and Matrix Setup

Coverage is estimated by dividing the sensing field into uniform grid areas. The number of sensors that cover each point on the grid is computed by calculating the euclidean distance between each grid point and the individual sensor's point using their coordinates. If the euclidean distance between the two points is within the sensing range R_s ; the point is taken to be covered by the sensor. As shown in Figure 5.2, some points can be covered by one or more sensors.

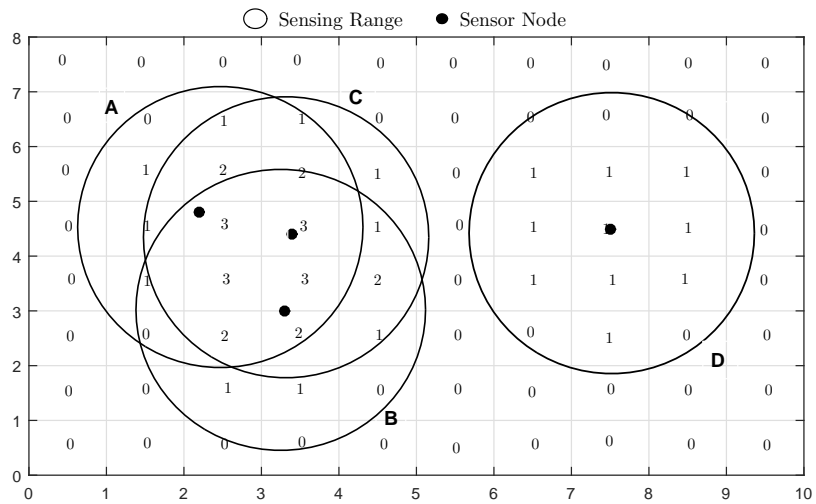


Figure 5.2: Covered Grid points in a 10×8 Sensing field

Inactive Node Selection

Conclusions as to which nodes to send into inactive mode at the beginning of each network operation round is made by the SSIN. The sleeping nodes candidate list evolves through the inspection of which nodes have residual energy less than the computed average energy. This selection process is tantamount to the Boltzmann selection process whereby a method is adopted to control the selection pressure [Dumitrescu et al. \[2000\]](#). The temperature parameter is varied in the Boltzmann selection process to effectively control the selection pressure. The maximum coverage effect, Max_{eff} is employed in the sleep scheduling mechanism to regulate the effect of putting sensors to sleep and is defined as:

$$Max_{eff} = 2 \times \pi \times R_s^2 \quad (5.7)$$

Here, R_s is the range over which a sensor node senses (taking the coverage area as a circle with radius R_s), $(\pi \times R_s^2)$ is the coverage of one node and the value '2' represents coverage of two nodes.

The coverage effect C_{eff} as shown in Figure 5.3, is the effect of putting a node to sleep based on coverage. The total coverage effect is computed by summoning a matrix called the Coverage Matrix. This matrix captures node coverage areas that overlap permitting the identification of nodes that can be placed into sleep mode without harming coverage as there will be other nodes covering the selected

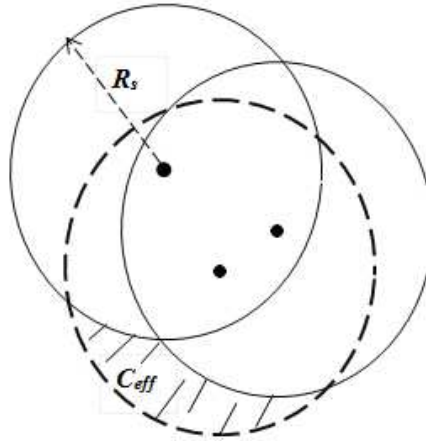


Figure 5.3: Illustration of Nodes to Sleep on Coverage Area

5. Proposed Dynamic Local Search-Based Algorithm

Algorithm 4 Proposed SSIN protocol

```

 $Acc_{eff} = 0;$ 
Compute the residual energy,  $E_{Res}$  of each node. (refer to section 2.1.3)
Compute the average energy of all nodes,  $E_{Avg}$ .
Generate a candidate list for nodes with  $E_{Res} < E_{Avg}$ .
Compute  $Max_{eff}$ . (refer to equation 5.7)
while ( $Acc_{eff} < Max_{eff}$ ) do
    Compute probability,  $P$  of adding nodes to the sleeping list. (See equation 5.8)
    if ( $rand() < P$ ) then
        Create list of sleeping node from the candidate list.
        Compute the coverage effect,  $C_{eff}$ .
         $Acc_{eff} = Acc_{eff} + C_{eff}$ 
    end if
end while

```

node's area. The accumulated Coverage effect Acc_{eff} is defined as the total effect on the coverage as a result of allowing some nodes to sleep. The SSIN mechanism presented here has been created to ensure the Acc_{eff} value is expected to be less than the Max_{eff} for optimum coverage ($Acc_{eff} < Max_{eff}$). The probability that a node will be added to the sleeping node list can be computed using:

$$P = e^{(-C_{eff}/Max_{eff})/(1-(Acc_{eff}/Max_{eff}))^2} \quad (5.8)$$

A randomly generated number is compared with the computed probability, P . A candidate list of inactive nodes is created if the random number is less than the probability, P . The accumulated frequency Acc_{eff} is computed by adding its current value to coverage effect C_{eff} value. The operations of SSIN continues until the Acc_{eff} is greater than the maximum acceptable coverage effect, Max_{eff} as shown in Algorithm 4.

5.4.2 Proposed ILSACHS protocol

Clustering is an efficient way in which a WSN can balance its load, save energy and enhance the network lifetime. It is the grouping of sensor nodes into clusters and CH selection for all the clusters. The CH plays a vital role of gathering data

5. Proposed Dynamic Local Search-Based Algorithm

Algorithm 5 Proposed ILSACHS protocol

```
i ← 0, searchlength = 100;
Generate a initial solution,  $s^*$ .
Compute the fitness value,  $F(s^*)$ .
while ( $i \leq searchlength$ ) do
    Mutate  $s^*$  to obtain a point  $s'$  using the perturbation operator (See Section
    5.4.2 for details).
    Combine the solution  $s^*$  with the point  $s'$  to obtain a new solution  $s^{*'}.$ 
    (refer to Algorithm 8)
    Compute the fitness value,  $F(s^{*'})$ .
    if ( $F(s^{*'}) < F(s^*)$ ) then
        Replace solution  $s^*$  with the new solution  $s^{*'}$ .
    else
         $i = i + 1$ 
    end if
end while
```

from its associated nodes and forwarding the aggregated data to the sink for processing. In the proposed ILSACHS protocol, the perturbation operator is used to generate a new starting point for further local searches for the local optimal solution Zhang and Sun [2006]. One of the major contributions of this thesis is that the mutation rate of the perturbation operator changes dynamically according to a predetermined step size in order to search outside the local optimum. Using the combination operator, the local optimal solution is mated with the local solution in the same neighboring area, and a new solution is produced (See Algorithm 5).

Iterated Local Search with Perturbation Operator

In the proposed ILSACHS protocol, the iterated local Search algorithm improves a solution in the search space by starting from an initial random solution $s^* \in S^*$, and iteratively explores the search space for a local optimal solution. The fitness value $F(s^*)$ of the solution s^* is accessed using the proposed objective function in Equation 5.4. At the first step, the local solution s^* is mutated by a perturbation operator to generate an intermediate solution or point s' . The current solution s^* is combined with point s' , and a new solution $s^{*'}$ (See Section 5.4.2) is produced with fitness value $F(s^{*'})$. If the fitness value $F(s^{*'}$ is greater than $F(s^*)$, a move

5. Proposed Dynamic Local Search-Based Algorithm

from solution s^* to $s^{*'} is accepted i.e. $s^{*'}$ replaces s^* . If the condition is not satisfied, solution $s^{*'}$ is discarded i.e. the current solution s^* remains unchanged.$

In the case where local search failed to move to a new solution after some consecutive number of search attempts within the predetermined search length, the mutation rate of the perturbation operator is increased. This action of perturbation operators is similar to the term *kickers* used in special purpose local searches for intensification or diversification. It allows the search to escape from the attraction area of a local minimum Shen et al. [2001]. The search length is defined as the total number of search attempts for iterated local search operator. As shown in Figure 5.4, the search length is divided into four step sizes which denotes the number of search attempts before the mutation rate is increased to widen the search area.

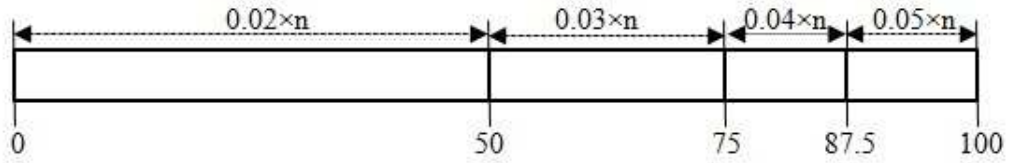


Figure 5.4: Step Size division of Search Length

Combination Operator

The point s' obtained by the perturbation operator is combined with the local solution s^* in order to obtain a new solution using a combination operator that uses heuristic crossover. The pioneer of this heuristic crossover operator is Lixin Tang Lixin [1999], and proposed to utilise parents' implicit information to produce offspring. In the canonical crossover approach, parents mate to produce pairs of offspring that tend to substitute their parents with no guarantee that an offspring produced would be better than either of its mating parent Hasan et al. [2007]. The heuristic combination approach uses a special crossover that has a knowledge of a problem to combine two candidate solutions to produce an improved solution.

The solution produced by this heuristic combination operator represents CH configurations that are well distributed across the sensing field and favors those

5. Proposed Dynamic Local Search-Based Algorithm

Algorithm 6 Proposed Combination Operator

Select the two solutions $s^{*'}$ and s^* .
Compute and store the CH position in solution $s^{*'}$ and s^* into set r_a and r_b respectively.
Compute the threshold distance, T (refer to Section 5.4.2)
Compute the union set $r_{a,b}$. (refer to Section 5.4.2)
Obtain the first cluster position $r_{a,b}(1)$ in the set $r_{a,b}$.
Create a new set r_{new} and transfer the $r_{a,b}(1)$ to it.
Compute the distance, D between CH positions in the sets r_a and r_b .
while ($D < T$) **do**
 if (Energy in $CH_{a,b}$ node is less than r_{new} node) **then**
 Discard the cluster head node. (i.e. do not add to r_{new} set)
 end if
 Replace the cluster head node in the r_{new} set
end while
Add to the cluster head node in the set $r_{a,b}$ into the r_{new} set.

with higher energy. The proposed heuristic crossover prohibits the selection of two CHs within the same region and higher priority is given to a CH with higher energy. The local solution $s^{*'}$ and perturb point s^* are selected from the iterative local search process and the CH position in the two solutions is computed and stored into the set r_a and r_b respectively. A threshold distance T is defined between two neighboring CH positions as; $T = \frac{\sqrt{(x_{max}-x_{min})^2+(y_{max}-y_{min})^2}}{n \times 0.04}$, where the (x_{min}, y_{min}) and (x_{max}, y_{max}) are minimum and maximum xy-coordinates of the sensor fields respectively, $(n \times 0.04)$ represent 4% of the total number of sensor nodes. The union of r_a and r_b is represented by $r_{a,b} = r_a \cup r_b$. By default, the first CH position $r_{a,b}(1)$ in the set $r_{a,b}$ is transferred to a newly created set r_{new} . Each subsequent CH position in the $r_{a,b}$ is compared with the r_{new} array set in order to make certain decisions which is based on distance between the CHs and their residual energy.

5.5 Energy Consumption Computation

In this thesis, the energy consumption for all the proposed algorithm is performed at the sink which is usually a device with unlimited energy. Energy is consumed

5. Proposed Dynamic Local Search-Based Algorithm

in each round of WSNs operations, and this operation phase is divided into the set-up and steady phase.

5.5.1 Set-up Phase

The set-up phase involves the transmission and reception of control packets k_{CP} from the sink to all nodes to initialise inter- and intra-communications. In set-up phase, the optimum number of clusters is found and clusters are created as shown in Figure 5.5. In first round, the sink sends a short message to wake up and to request the IDs, positions and energy levels of all sensor nodes in the sensor field. Based on the feedback information from sensor nodes, the sink uses the proposed clustering protocols to find the optimum number of CHs and their locations based on minimization of the dissipated energy on communication process as shown by shaded block in Figure 5.5. Also, the sink assigns members nodes for each CH. Once CHs are selected and members of CH are assigned, the sink sends a short message to inform each CH by IDs of its member nodes then send a short message that contains CH's ID to all member nodes to inform each member node the CH to join. Based on a short message received from the sink, each CH creates the TDMA schedule by assigning slots to its member nodes and informs these nodes by this schedule. The TDMA schedule is used to avoid intra-cluster collisions and reduce energy consumption between data messages in the cluster and enables each member of the radio equipment off when not in use. The details of the proposed clustering protocol is shown in 5.5.

Similar to Equation 5.2, the energy $E_{Rx}(k_{CP})$ is spent by each sensor to receive the control packets from the sink. All sensors report their IDs, positions and energy levels back to the sink and the transmitted energy $E_{Tx}(k_{CP}, d)$ consumed for the task is similar to Equation 5.1. The control packet received from all sensor nodes is processed by sink to make the following vital decisions; which nodes to keep active, CH selection, and the associated CH membership. Also, considered the energy $E_{Rx}(k_{CP})$ dissipated in receiving the membership status information from the sink. Elected CHs are required to transmit a TDMA schedules to their respective members and the energy dissipated to perform the task is computed

5. Proposed Dynamic Local Search-Based Algorithm

using:

$$E_{Tx(ch_i)}(k_{CP}, d_{i-toMem}) = \sum_{i=1} ch_i \times E_{Tx}(k_{CP}, d_{i-toMem}) \quad (5.9)$$

Additionally, each member node also spends energy $E_{Rx}(k_{CP})$ to receive TDMA schedule from the CHs

5.5.2 Steady State Phase

During the steady state phase, the active sensor nodes begin sending data packets (k). Each node sends the sensed data to its CH according to the TDMA schedule received. The CH node receiver must always be ready to receive packets from its nodes within its cluster. Data aggregation is performed on all received data at the CHs and all data are converted into a single data stream. This aggregated data stream is transmitted from the CHs to the sink. This process consumed some amount of energy by the sensor node transceiver as Equation 5.12. The total amount of energy spent by all member nodes to transmit to their respective CHs is computed using:

$$E_{Tx(m_i)}(k) = \sum_{i=1}^p m_i \times E_{Tx}(k, d) \quad (5.10)$$

Assume number of members p , then $m_i = 1, 2, 3, \dots, p$ and $E_{Tx}(k, d)$ is the transmission energy that depends on the packets (k) and distance between each member and respective CH. And the total amount of energy dissipated by all CHs for receiving data packets from their member nodes is given as:

$$E_{Rx(m_i)}(k) = \sum_{i=1}^p m_i \times E_{Rx}(k) \quad (5.11)$$

Where $E_{Rx}(k)$ is the energy required for each CH to receive data packets from its members. Also, the energy dissipated by the CHs to aggregate the data received from all its members and itself can be calculated using:

$$E_{DA(m_i+1)}(k) = kE_{DA} * \left(\sum_{i=1}^p m_i + 1 \right) \quad (5.12)$$

5. Proposed Dynamic Local Search-Based Algorithm

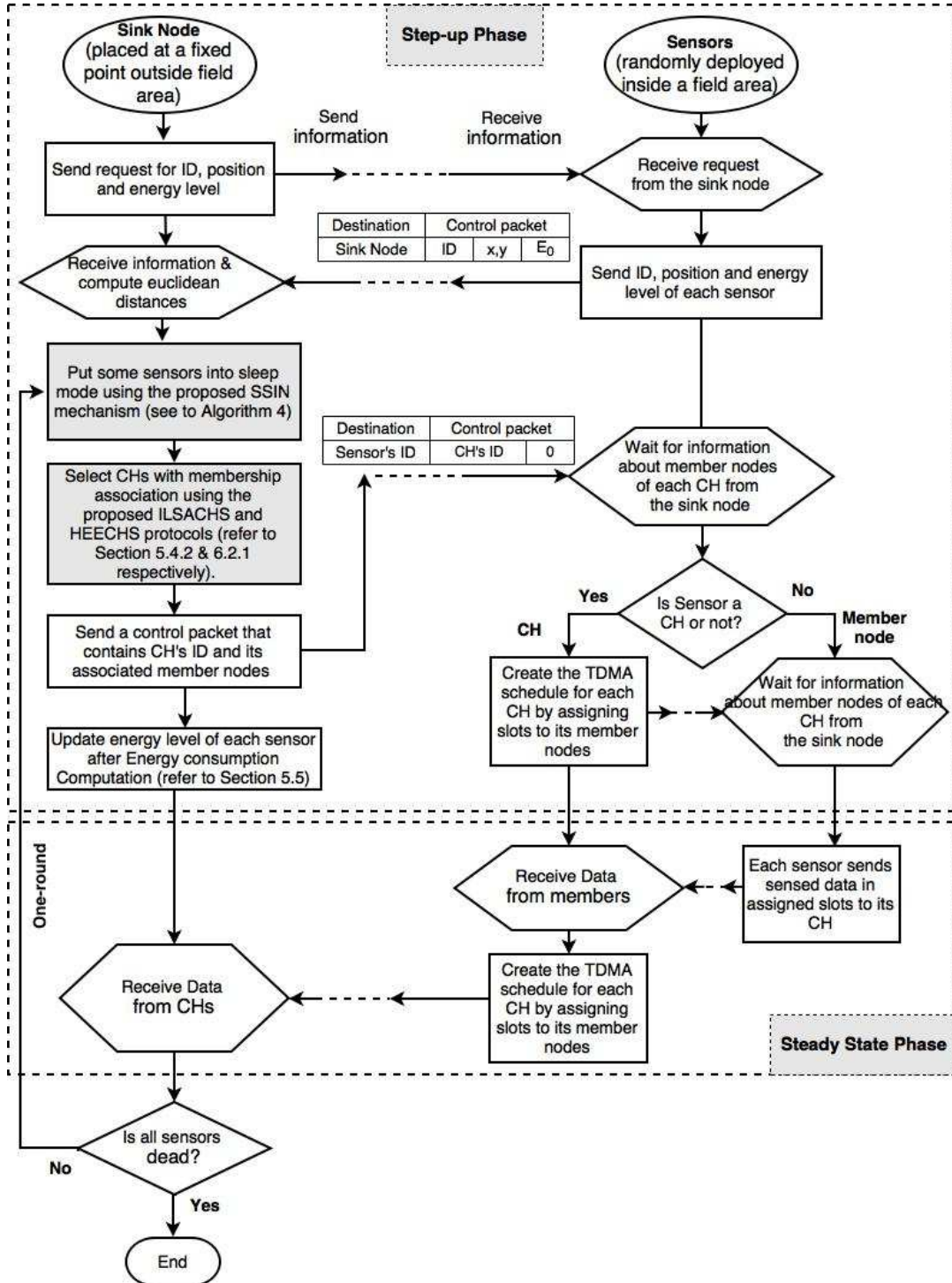


Figure 5.5: The operational sequence of the proposed clustering protocols

5. Proposed Dynamic Local Search-Based Algorithm

Where kE_{DA} is the energy required for CHs to aggregate (k) amount of data packets received from the members and CH itself. Finally, the amount of energy spent by the CHs node for transmitting data packets to the sink is computed using:

$$E_{Tx(ch_i)}(k, d_{i-toSink}) = \sum_{i=1}^q ch_i \times E_{Tx}(k, d_{i-toSink}) \quad (5.13)$$

Assume number of cluster-heads (q) in the overall network, then $ch_i = 1, 2, 3, \dots, q$ and $E_{Tx}(k, d_{i-toSink})$ is the transmission energy that depends on the packets (k) and distance between each CH and the sink ($d_{i-toSink}$). Thus, the total energy consumed by all the CHs can be computed using:

$$\begin{aligned} E_{CHs} = & 2 * E_{Rx}(k_{CP}) + E_{Tx}(k_{CP}, d_{i-toSink}) + E_{Tx(ch_i)}(k_{CP}, d_{i-toMem}) \\ & + E_{Rx(m_1)}(k) + E_{DA(m_i+1)}(k) + E_{Tx(ch_i)}(k, d_{i-toSink}) \end{aligned} \quad (5.14)$$

And the energy dissipated by all the member nodes is computed as:

$$\begin{aligned} E_{Mem} = & E_{Tx}(k_{CP}, d_{i-toSink}) + E_{Tx}(k_{CP}, d_{i-toCH}) \\ & + 3 * E_{Rx}(k_{CP}) + E_{Tx(m_i)}(k) \end{aligned} \quad (5.15)$$

Therefore, the overall energy dissipated by all nodes is represented by $E_{TOTAL} = E_{CHs} + E_{Mem}$. Also, note that the residual energy of each node (either a CH or member node) at each round is updated by subtracting the energy consumption from the current residual energy.

5.6 Performance Evaluation

This work evaluate the protocols from an energy efficiency perspective by examining the number of alive nodes versus rounds. The graphed results helps us to evaluate the lifetime of the sensor nodes using the proposed algorithm. The simulation models and programs are developed using the MATLAB tool. From my point of view the proposed technique is scalable and may lead to energy efficiency

5. Proposed Dynamic Local Search-Based Algorithm

improvement in different network sizes. To assess this claim the performance of DLSACH is compared to three other protocols LEACH, TCAC and SEECH in five experiments; first three experiments are homogeneous networks and last two experiments are heterogeneous networks. Table 5.2 describes the parameter values for each experiment in details. The common communication parameters used for all the experiments presented in Table 5.2 are listed in Table 5.1.

Table 5.1: Communication Parameters with Specified Values

Parameter	Value
Electronics Energy, E_{elect}	50nJ/bit
Multipath Loss, ε_{mp}	0.0013pJ/bit/ m^4
Free space Loss, ε_{fs}	10pJ/bit/ m^2
Aggregation Energy, E_{DA}	5nJ/bit/signal
Threshold Distance, d_0	87m
Control Packet size, k_{CP}	50
Packets size, k	400

Table 5.2: Parameter values for each experiment

Experiments	Parameter Settings		
	Sink Coordinates (m)	Number of Sensors	Initial Energy (J)
Experiment I	(50,175)	100	0.5
Experiment II	(50,200)	400	0.5
Experiment III	(50,50)	100	0.5
Experiment IV	(50,50)	100	$\mu=0.5$ (All Nodes) $\sigma=0.05$
Experiment V	(50,50)	100	$\mu=0.5$ (75 Nodes) $\sigma=0.05$ 0.5J (25 Nodes)

Note: μ and σ represent the mean and standard deviation of the sensor node energy distribution in Experiment IV & V.

5.6.1 Performance Measures

There are many metrics used to evaluate the performance of the clustering protocols [Lixin \[1999\]](#). These measures are used in this thesis to evaluate the performance of DLSACH protocol:

1. **First Dead Node (FDN)**: This is the number of rounds at which the first node dies (FND). It can also be referred to as the operational lifetime or stability period of the network. Therefore, larger FND value signifies longer WSN stability periods.
2. **Last Dead Node (LDN)**: This is the number of rounds from the start of network operation until the last node dies (LND).
3. **Instability Period Length (IPL)**: The round difference between the round at which the last node dies and the first node dies (i.e. $IPL = LND - FND$).
4. **Average Energy at first node dies (AEFND)**: This is a new performance measure proposed in this work to evaluate the average energy of all sensor nodes when the first node dies.

Clearly, the longer the stability period and the shorten the instability period are, the better the reliability of the clustering process of the WSN.

5.6.2 Results and Discussion

The average value of performance measures are obtained from 100 simulation runs and presented in this section for analysis. For each simulation run, new sensor node are distributed in a sensor field area. The proposed DLSACH protocol is compared with LEACH, TCAC and SEECH for a small and large scale network of 100 and 400 sensor nodes respectively and is shown in [Figure 5.6](#), which depicts the number of alive nodes during simulation time versus the number of rounds.

Also, the FND, LND and IPL values belonging to the graphs in [Figure 5.6](#) are presented in [Table 5.4](#). The FND values presented in [Table 5.4](#) shows that the DLSACH protocol maintains the network operational lifetime of 46, 141 and 348

5. Proposed Dynamic Local Search-Based Algorithm

more than the SEECH, TCAC and LEACH respectively for Experiment I (100 nodes). In Experiment II, the FND value of DLSACH is higher compared with the other three protocols. This result of the experimentation shows that the energy of the sensor nodes is balanced and extend for a longer period. When the FND time is reached, most of the nodes begin to die due to insufficient energy and this is represented by a sharp decline in the slope of TCAC, SEECH and DLSACH as shown in Figure 5.6. Also, figure 5.6 indicates a late decline in line graph for the proposed DLSACH protocol, which means that the round at which the first node dies is longer than the other three protocols.

For Experiment II, the instability period for the proposed DLSACH is 50, 49 and

Table 5.3: Comparison of LEACH, TCAC, SEECH and DLSACH for FND,LND and IPL

Experiment	Protocol	Performance Measure (Round)		
		FND	LND	IPL
Experiment I (100 Nodes)	LEACH	726	1209	483
	TCAC	933	1006	73
	SEECH	1028	1099	71
	DLSACH	1074	1166	92
Experiment II (400 Nodes)	LEACH	685	1274	589
	TCAC	948	1071	123
	SEECH	1016	1140	124
	DLSACH	1206	1280	74
Experiment III (1000 Nodes)	LEACH	672	2014	1342
	TCAC	725	1664	939
	SEECH	1587	2202	615
	DLSACH	183	674	491

Table 5.4: Performance Measures for Experiment III,IV and V

Performance Measures	Experiments		
	Experiment III	Experiment IV	Experiment V
FND	1354	1246	1241
LND	1445	1336	1331
IPL	91	90	90
AEFND	0.014949	0.014518	0.016568

5. Proposed Dynamic Local Search-Based Algorithm

515 less than the SEECH, TCAC and LEACH. This low value reveals that the proposed protocol performs better in dense network. The results of three experiments which consist of one homogeneous (Experiment III) and two heterogeneous (Experiment IV & V) sensor networks is presented in Table 5.4. Experiment III has a higher FND values of 1354 than Experiment IV & V, which are 1246 and 1241 respectively. This shows that the higher the complexity of the problem, the lower the FND value. Two interesting points from experiment I & II is that sen-

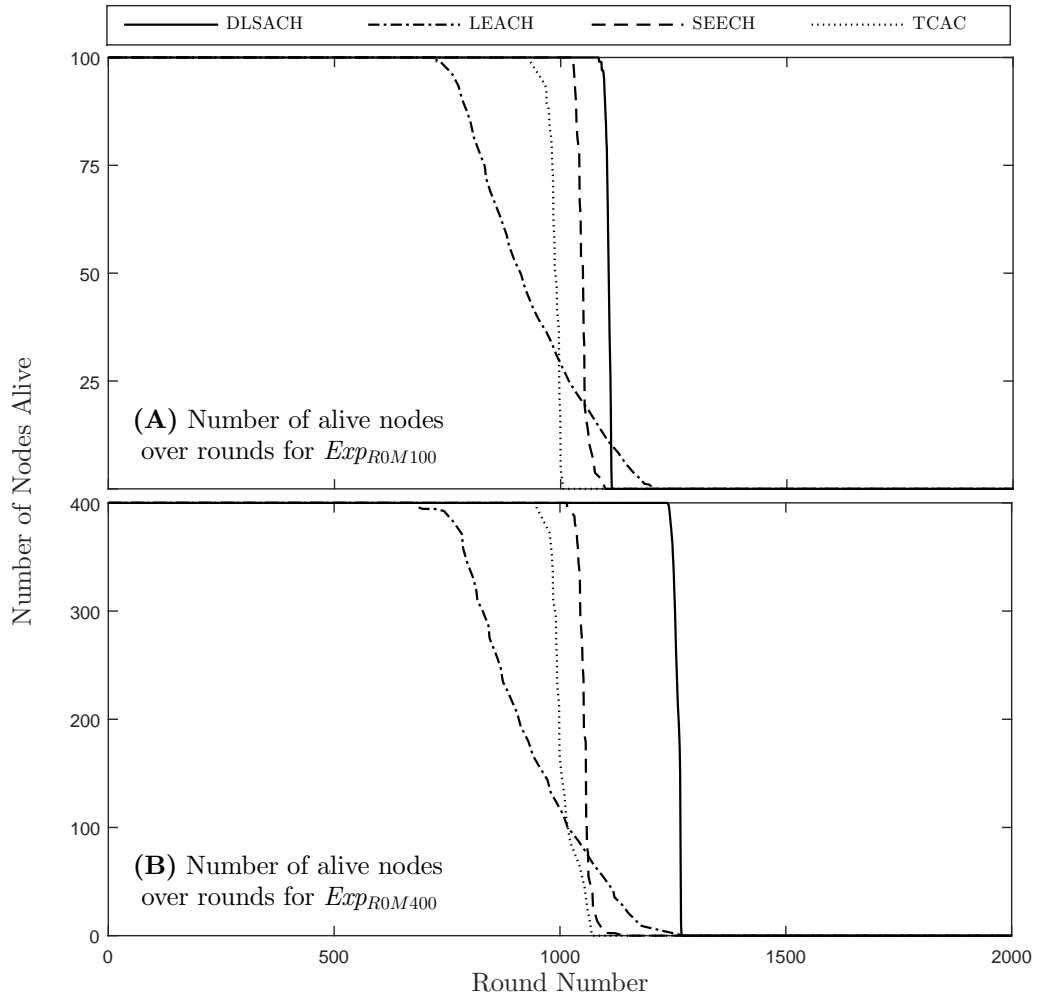


Figure 5.6: Network Lifetime Comparison of DLSACH with LEACH, SEECH, TCAC

5. Proposed Dynamic Local Search-Based Algorithm

sors were able to keep the instability period almost constant for all the last three experiments and also keep the AEFND value very low number; which means almost all the sensor's initial energy were used till the death of first node. However, the proposed DLSACH protocol shows a very poor performance at large network size of 1000 sensors as seen in Experiment III result compared to other protocols.

It can therefore be deduced that the proposed DLSACH protocol decreases energy consumption and optimises energy balancing, thus increasing the network lifetime with the exception of dense and large network size. At each round, the proposed DLSACH protocol conserves energy by selectively allowing some nodes to become inactive before network operation. Also, it uses an Iterated Local Search Algorithm to select the best CHs configuration ensuring that CHs are well distributed around the sensor field.

5.7 Conclusion

In this chapter, Dynamic Local Search-Based Algorithm for Clustering Hierarchy (DLSACH) protocol was proposed for energy management in wireless sensor networks. The two major operations in this protocol include sending some nodes into sleep mode and the cluster head selection. The Iterated Local Search Algorithm for Cluster Head Selection (ILSACHS), a mechanism that employs iterative local search with perturbation operator is proposed for solving cluster head selection problem. The perturbation operator mutates a random solution to produces a point within the local optimum. This point is combined with the selected random solution to produce a new solution using a heuristic based combination operator. A move to the new solution is accepted only at a higher fitness value, otherwise the move is discarded. After a complete search process, the new solution that guarantees optimally distributed cluster heads is applied to the network. A new mechanism called the Stochastic Selection of Inactive Node (SSIN) that is inspired by Boltzmann selection process is proposed to stochastically select which nodes to send into sleep mode without adversely affecting coverage. The two proposed mechanisms work collaboratively to reduce and balance energy consumption by selecting well distributed nodes with higher energy as cluster heads in order to prolong network lifetime. Results shows that the network lifetime of DLSACH

5. Proposed Dynamic Local Search-Based Algorithm

protocol is more than SEECH, TCAC and LEACH protocol for network sizes of 100 and 400 number of sensors. Also, the proposed **DLSACH** protocol shows good performance for heterogeneous sensor networks in terms of energy consumption and stability periods. However, **DLSACH** protocol cannot perform well under a large network size (1000 number of sensors) compared with other protocols. A dynamic global-based search strategies that employs a novel heuristic crossover to obtain a better solution that efficiently use the energy supply constraints of battery-powered sensors to prolong its network lifetime is proposed in the next chapter.

Chapter 6

Proposed Global-based Search Algorithm

This chapter proposes a novel Heuristic Algorithm for Clustering Hierarchy (**HACH**), which sequentially performs selection of inactive nodes and cluster head nodes at every round. Inactive node selection employs a stochastic sleep scheduling mechanism to determine the selection of nodes that can be put into sleep mode without adversely affecting network coverage. Also, the clustering algorithm uses a novel heuristic crossover operator to combine two different solutions to achieve an improved solution that enhances the distribution of cluster head nodes and coordinates energy consumption in WSNs. The proposed algorithm is evaluated via simulation experiments and compared with some existing algorithms. **HACH** protocol shows improved performance in terms of extended lifetime and maintains favourable performances even under different energy heterogeneity settings.

6.1 Introduction

The problem of CH selection can be considered as an optimisation issue where the methods have employed GA to solve. This chapter define an objective function that evaluates the discrete solution and propose an innovative heuristic crossover which is enhanced by the knowledge of problem in view. This thesis present a new Heuristic Algorithm for Clustering Hierarchy (**HACH**) protocol that simulta-

6. Proposed Global-based Search Algorithm

neously performs sleeping scheduling and clustering of sensor nodes upon each round. For sleep scheduling operation, this chapter present the stochastic selection of inactive nodes (SSIN). A protocol that imitates the Boltzmann selection process in GA was used to decrease the number of active nodes in each round by putting some nodes to sleep or into inactive mode so that energy could be conserved and network lifetime increased without harming coverage. Furthermore, the Heuristic-Crossover Enhanced Evolutionary Algorithm is developed for Cluster Head Selection (HEECHS) protocol for the clustering operation. HEECHS uses the known information around the problem to develop a useful heuristic crossover that combines genetic material in a unique way to produce improved CH configuration. This method described has some parallels with optimisation algorithms known as Memetic Algorithms (MAs). This algorithm is a type of stochastic global search heuristics in which Evolutionary Algorithm-based techniques are mixed with a local search technique to improve the quality of the solutions proposed by evolution Hart et al. [2005]. Sleep scheduling and clustering algorithms work together to optimise network lifetime by harmonising energy consumption amongst sensor nodes during the communication times. Energy consumption optimisation is performed by selecting spatially distributed nodes with higher energy as CHs and additionally placing certain nodes into sleep mode without harming coverage. The HACH protocol proposed performs very well compared to protocols that use GA because it integrates knowledge of the problem into GA crossover operator.

The rest of this chapter is organised as follows. In Section 6.2, the proposed algorithm under three pivotal operational phases, those being the sleep scheduling mechanism, clustering algorithm and the energy consumption calculation is discussed. Section 6.3 presents the performance evaluation in terms of the stability period and network lifetime, average energy at first node dies (AEFND) and heterogeneity measure. Finally, Section 6.4 provided the conclusion.

6.2 The proposed HACH Protocol

There are three consecutive operations within the proposed protocol: sleep scheduling, clustering and network operations. The network and radio assumption dis-

6. Proposed Global-based Search Algorithm

Algorithm 7 HACH Protocol

Let $AliveNodes$ be the total number of sensor nodes
Compute the network total coverage.
while ($AliveNodes > 0$) **do**
 Use algorithm SSIN to select inactive nodes. (See Algorithm 4)
 Put selected nodes into sleep mode.
 Apply the proposed HEECHS algorithm for CHs configuration. (See Algorithm 8)
 Compute the energy values of E_{CH} , E_{Mem} and E_{Res} . (refer to Section 2.1.3)
 Find out the number of dead node (node with energy equal or less than 0).
 Update $AliveNodes$
end while

cussed in section 5.2 of chapter 5 is applicable to this protocol. The sink transmits control packets at the initial set-up phase so that it can receive node information in terms of the nodes ID, location and energy. During sleep scheduling operations, the SSIN protocol proposed dynamically selects the nodes to send to sleep by generating an initial candidate list. This list is populated with nodes that have lower energies than the average energy of all nodes. Employing a stochastic process, a small number of nodes are subsequently placed into sleep mode without harming coverage (See Section 5.4.1 for details). CH selection employing HEECHS is then completed on the remaining active nodes. At the end of each round, the energy consumed by CHs (E_{CH}), member nodes (E_{Mem}) and residual energy (E_{Res}) of each sensor are computed (refer to section 5.5 for details). The residual energy obtained is used as a parameter to make inactive node and CH selection decision at the beginning of the next round. Sensor dies when the residual energy is equal or less than zero and the number of sensors alive is updated at the end of every round. The network cycle continues until the number of nodes alive is equal to zero.

The proposed HEECHS protocol operates at the network layer of WSNs layered model presented in Charfi et al. [2009], which is similar to the Open System Interconnection (OSI) network model. After nodes deployment, the sink transmits and receives control packets containing the coordinates and energy value of all nodes. Using the obtained sensor coordinates, the sink computes the Eu-

6. Proposed Global-based Search Algorithm

clidean distances between two adjacent nodes and each node to the sink. These Euclidean distances and energy values are both used in establishing the cluster-based network topology for the purpose of packet routing.

Here, clustering can be considered as an optimisation problem which can best accomplished using GA. Tournament selection, mutation operator and the heuristic crossover are the genetic operators used in this approach. The most suitable CH configuration which guarantees balanced energy consumption across the network topology is selected at every network operation round. The residual energy of each node is calculated at the end of each round. This computed value is then employed to calculate the average energy for the next round. This cycle subsequently repeats until all network nodes are dead, as shown in Algorithm 7.

6.2.1 Clustering Operations using HEECHS protocol

The clustering operation is divided into stages: CH selection, cluster formation, data aggregation and data communication. As shown in Figure 6.1, the setup state starts by the CH selection stage and proceeds by cluster formation. The setup state is followed by the data transmission state, which is subdivided into data aggregation and data transmission phases. During the setup state, a sink-assisted clustering algorithm that performs CH selection and membership association is applied to the active nodes in the network. During network initialization, sensors send their energy and location information to the sink in order to implement the proposed algorithm. The HEECHS protocol favours the selection of a CH

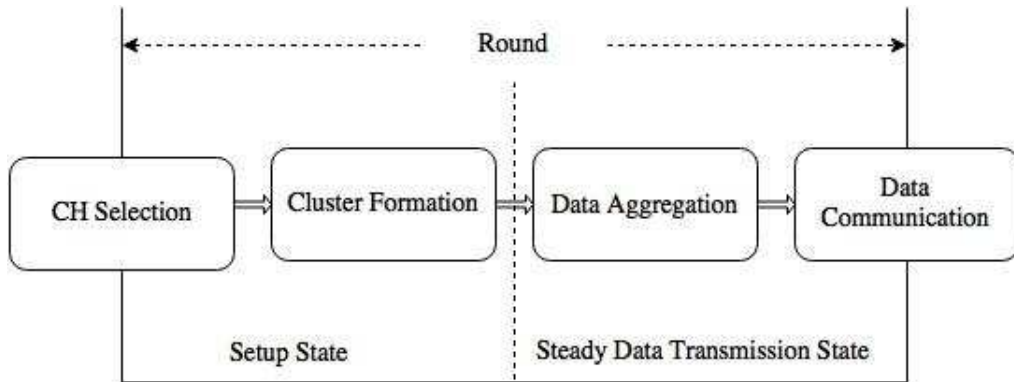


Figure 6.1: One round of the clustering process

6. Proposed Global-based Search Algorithm

that has higher energy and is far from neighbouring CH. As illustrated in Figure 6.2, an energy efficient sensor node distribution is constructed by the proposed HACH protocol at every network operation round. Sensors are assigned to the closest CH thereby forming a single cluster. A TDMA schedule is assigned for each cluster to schedule packet transmission to that CH by the member nodes. All the information about clusters and TDMA schedule packets is broadcast to the entire network. Based on the time slot in the TDMA schedule packets, each node in a cluster send sensed data to their respective CH.

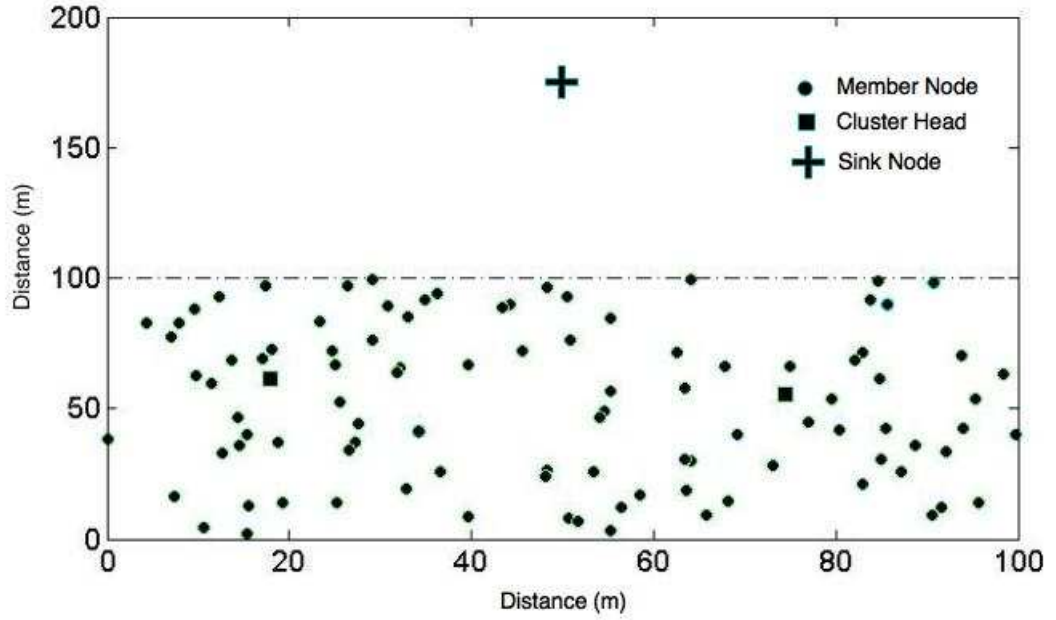


Figure 6.2: Sensor nodes Topology and Random distribution

At each round, the sink performs a re-clustering procedure to form a new cluster-based topology that preserves the WSNs coverage and energy efficiency characteristics by rotating the CH role among sensors with scalability of hundreds to thousands. Scalability implies that there is a need for balanced energy consumption among the sensor nodes during communication through an efficient clustering algorithm Mamun [2012]. The CH loses energy faster than the member nodes; hence the need for re-clustering or rotating the CH role among sensors in order to balance the energy consumption. Re-clustering is performed at the end of a round, which is the total time span for a processes involved in the setup and

6. Proposed Global-based Search Algorithm

steady data transmission state. The time-length of each round must be carefully decided because a large time length drains CHs energy and a short time-length result into overhead caused by frequent re-clustering [Pal et al. \[2013\]](#). The round time-length of the proposed algorithm adjust itself dynamically based on the number of active nodes in the WSNs.

In this work, the HEECHS protocol proposed is developed for the CH selection task using a heuristic-based GA. It runs through a number of tasks, similar to conventional GAs, such as population strings creation, string evaluation, best string selection and finally reproduction to create a new population. The unique, but significant difference is that the HEECHS protocol employs a problem-dependent knowledge-based heuristic crossover to find the best CH configuration with the optimum number of appropriately distributed CH nodes. In the proposed HEECHS, the genetic process of finding the best solution is performed using an energy unlimited sink device that can handle high execution time complexity and computation. The individuals within population $P(t)$ are coded by 0-1 binary representation where '0' denotes a member node and '1' denotes a CH node as shown in figure [6.3](#).

Each individual with length N_s in a population size p_s is evaluated by computing the fitness value using Equation [5.4](#). Individuals with the best fitness value are selected from two randomly selected parent pairs, $P(x)$ and $P(y)$. This process continues until the mating pool is filled. The heuristic crossover proposed here is subsequently applied to the individuals in the pool and a new population $P(t+1)$ is produced. Again, each individual fitness value in this new population is computed using Equation [5.4](#) and the entire cycle continues until the stopping criterion is achieved. The stopping criterion is realized when the populations average fitness undergoes no further changes.

6.2.2 Proposed Heuristic Crossover

The principal operator used in the HEECHS protocol to produce new solutions is the heuristic crossover. This is a problem-dependent crossover that utilises knowledge of a problem to fuse two potential resolutions, producing a new solution. According to Lixin Tang [Lixin \[1999\]](#), a heuristic crossover is an operator that

6. Proposed Global-based Search Algorithm

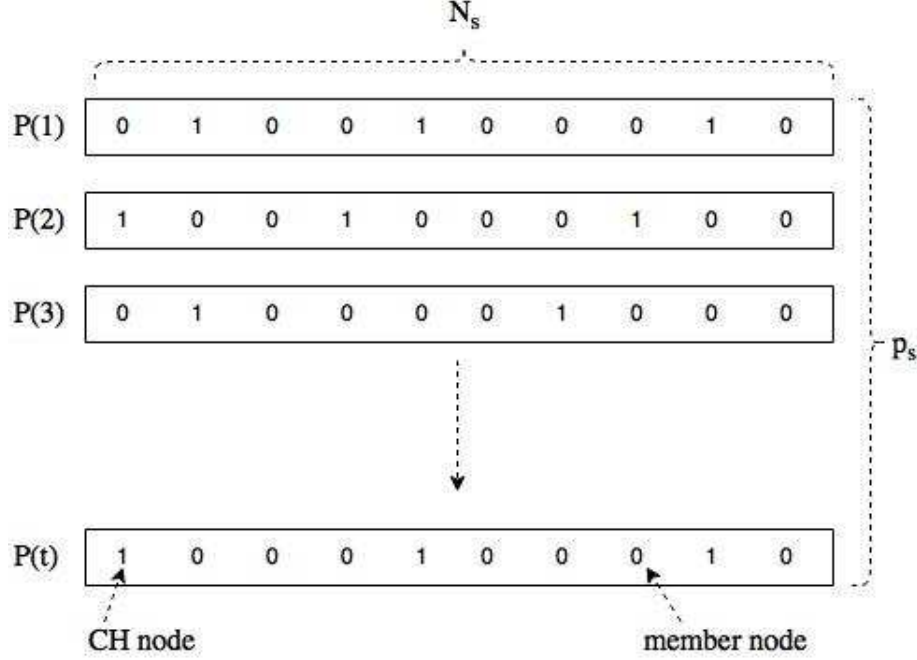


Figure 6.3: Binary representation of individuals in the population

makes use of parents' inherent information to produce an offspring. In the canonical approach, individuals in a population are selected and two parent individuals are combined using the crossover operator to produce a pair of offspring that will replace its parents. Correspondingly, there is no assurance that an offspring would be superior to its parents in the canonical approach Hasan et al. [2007]. Contrarily, the heuristic crossover operator generates only one offspring from two or more parents and it is certain that the offspring would be of higher quality than the parents. As shown in Algorithm 8, the proposed heuristic crossover generates a single solution with CHs that are spatially distributed in the sensor field and selects nodes with higher energy to be the CH.

The CH genes position in each individual of selected parent pair is computed. An array that holds the genes position in both parent pairs is expressed by CH_1 and CH_2 . Also, the threshold distance between any two adjacent CH position is defined as $\frac{\sqrt{(x_{max}-x_{min})^2+(y_{max}-y_{min})^2}}{n \times 0.04}$, where the (x_{min}, y_{min}) and (x_{max}, y_{max}) coordinates represent the minimum and maximum xy points in the sensing field, $(n \times 0.04)$ indicates 4% of all sensor nodes. A set CH_{all} is generated from the

6. Proposed Global-based Search Algorithm

Algorithm 8 Proposed Heuristic Crossover

Select two individuals from the parent population.
Compute and keep the CH position in each individual in CH_1 and CH_2 .
Compute the threshold distance, T (refer to Section 5.4.2)
Compute the union set $CH_{all} = CH_1 \cup CH_2$
Obtain the first CH position $CH_{all}(1)$ in the CH_{all} set.
Generate a new set CH_{new} and transfer the $CH_{all}(1)$ to it.
Compute the distance, D between CH positions in the sets CH_1 and CH_2 .
while ($D < T$) **do**
 if (CH_{all} node energy $<$ CH_{new} node energy) **then**
 Discard the CH node. (i.e. do not add to CH_{new} set)
 end if
 Replace the CH in the CH_{new} set
end while
Add to the CH in the set CH_{all} into the CH_{new} set.

union of CH1 and CH2 (refer to Algorithm 8). The first CH position in the union set CH_{all} is moved into a new set CH_{new} by default. As shown in Algorithm 8, the decision to move successive CH positions from the CH_{all} to CH_{new} is based on spatial distance between CHs and residual energy.

6.2.3 Other Operators

The efficacy of a genetic algorithm relies upon maintaining a balance between the concept of exploration and exploitation. Exploration is provided by crossover and mutation while selection enables exploitation Brunda et al. [2012]; Halke and Kulkarni [2012]. The rest of the operators used in the proposed HEECHS protocol are discussed below:

- The **Tournament selection operator** is a method of selecting an individual from a population of individuals in a genetic algorithm. Tournament selection involves selecting individuals with the best fitness from group of individuals randomly chosen from the current population. The selection pressure depends on the tournament size of the operator. In order to reduce the selection pressure, I used a tournament size of two for the proposed algorithm and this process continues until the mating pool is full.

6. Proposed Global-based Search Algorithm

- The **Mutation operator** is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. Mutation changes an individual (parent) with a mutation probability (m_p) to produces one individual (offspring) with new fitness value. The mutation probability indicate the number of bits in a genetic sequence that will changed from its original state. In the proposed algorithm, mutation operator with probability $m_p=0.1$ is applied each selected individual to obtain new individual with better fitness.

The parent and child individuals in the initial population pool produced in the previous step are arranged in ascending order based on their fitness value. Subsequently, individuals with minimum fitness values are selected and they form next generation's population. The *stopping criterion* is achieved when there is no further change in the fitness value of the population.

6.3 Performance Evaluation

The performance of clustering protocols can be evaluated using different types of metrics [Lixin \[1999\]](#). In this work, a MATLAB simulation model was developed to test the performance of proposed HACH protocol in terms of lifetime evaluation of sensor nodes. The experimental conditions for all of the trials investigated are presented in Tables [6.1](#) and [6.2](#). In each simulation run, the sensors under test are randomly redistributed in an (x, y) grid with origin (0, 0) and a deployment

Table 6.1: Parameter settings for Homogeneous WSNs Scenarios

Experiment	Parameter			
	Number of Sensors	Sink Coordinates (m)	Deployment Area (m ²)	Initial Energy (J)
Exp_{R0M100}	100	(50,175)	100×100	$\mu=0.5$ $\sigma_M=0$
Exp_{R0M400}	400	(50,200)	100×100	$\mu=0.5$ $\sigma_M=0$
$Exp_{R0M1000}$	1000	(50,350)	200×200	$\mu=1.0$ $\sigma_M=0$

6. Proposed Global-based Search Algorithm

Table 6.2: Parameter settings for Heterogeneous WSNs Scenarios

Experiments	Parameter				
	Number of Heterogeneous Nodes (R)	Number of Homogeneous Nodes (M)	Sink Coordinates (m)	Deployment Area (m ²)	Initial Energy (J)
<i>Exp_{R25M0}</i>	25	0	(50, 175)	100×100	$\mu=0.5$ $\sigma_R=0.05$
<i>Exp_{R50M0}</i>	50				
<i>Exp_{R75M0}</i>	75				
<i>Exp_{R100M0}</i>	100				
<i>Exp_{R25M75}</i>	25	75	(50, 175)	100×100	$\mu=0.5$ $\sigma_R=0.05$ $\sigma_M=0$
<i>Exp_{R50M50}</i>	50	50			
<i>Exp_{R75M25}</i>	75	25			

area of 100×100 m² or in the case of 1000 nodes over 200×200 m². Each trial has only one sink that is placed at a location outside the sensor deployment area with coordinates provided in Tables 6.1 and 6.2; the number of CHs is dynamic.

After clustering, the estimated maximum distances between a member node and a CH were found to be 39.20m, 29.43m and 26.17m for 100, 400 and 1000 sensor trials respectively. Also, the estimated maximum distance between a CH and the sink node were 126.55m, 141.82m and 303.42m for the 100, 400 and 1000 sensor trials respectively. The proposed HACH protocol is considered scalable in sense that it improves its energy efficiency as the network size increases. To demonstrate this fact, the performance of the proposed protocol was compared with SEECH, TCAC and SEECH protocols using experiments *Exp_{R0M100}*, *Exp_{R0M400}*, *Exp_{R0M1000}* which represent 100, 400 and 1000 homogeneous sensor nodes respectively and zero heterogeneous nodes in terms of initial energy value (refer to Table 6.1). Also, Table 6.2 presents experiment *Exp_{R25M0}*, *Exp_{R50M0}*, *Exp_{R75M0}*, *Exp_{R100M0}* which has 25, 50, 75, 100 heterogeneous sensor nodes respectively and no homogeneous nodes. Lastly, more experiments that mixed heterogeneous nodes with homogeneous nodes are conducted, namely experiments *Exp_{R25M75}*, *Exp_{R50M50}*, *Exp_{R75M25}*. The communication parameters used for all the experiments presented in Table 6.1 and 6.2 is shown in Table 5.1 of Section 5.

In addition to the simulation parameters in Table 5.1, the GA parameters are set as population size, $p_s=100$ and mutation rate, $p_m=0.05$. R and M signify the

6. Proposed Global-based Search Algorithm

number of heterogeneous and homogeneous sensor nodes respectively. In Table 6.1 and 6.2, μ represents the sensor nodes mean energy, σ_R and σ_M represent the standard deviation of heterogeneous and homogeneous sensor nodes respectively. For all experiments in Table 6.2, the mean initial energy E_0 used is 0.5J.

6.3.1 Stability Period and Network Lifetime

The stability period length (SPL) is the time range from the start of network operation until when the first node dies (FND) whereas the instability period (IPL) is the timespan from the FND until the last node dies (LND). The WSN lifetime is the time range from the start of network operation until the last node dies, which exclude energy unlimited sink devices. Immediately after the last sensor dies, the WSNs will stop its operation because the sink has lost its connectivity from the sensors. Alternatively, the WSNs lifetime can be defined as the combination of stability and the instability period. A reliable clustering process is characterised by a long SPL and a short IPL. Experimental results shown in Figure 6.4 depict the number of nodes that are alive after each round.

The performance of the proposed protocol is compared with other protocols

Table 6.3: Performance comparison of LEACH, TCAC and SEECH with HACH

Experiment	Protocol	Performance Measure (Round)		
		FND	LND	IPL
<i>Exp_{R0M100}</i> (100 Nodes)	LEACH	726	1209	483
	TCAC	933	1006	73
	SEECH	1028	1099	71
	HACH	1064	1167	103
<i>Exp_{R0M400}</i> (400 Nodes)	LEACH	685	1274	589
	TCAC	948	1071	123
	SEECH	1016	1140	124
	HACH	1235	1307	72
<i>Exp_{R0M1000}</i> (1000 Nodes)	LEACH	672	2014	1342
	TCAC	725	1664	939
	SEECH	1587	2202	615
	HACH	1789	2010	221

6. Proposed Global-based Search Algorithm

in terms of the FND, LND, and IPL measures as seen on the graphs presented in Figure 6.4. Table 6.3 shows that the proposed HACH protocol maintains the network operational lifetime of 338, 131 and 36 more than the LEACH, TCAC and SEECH respectively for Experiment Exp_{R0M100} . For a medium density WSN scenario Exp_{R0M400} , HACH shows a longer lifetime of 1235 rounds compared with LEACH, TCAC and SEECH which have a lower value of 685, 948 and 1016 respectively. The most fascinating result is that under the most dense WSNs

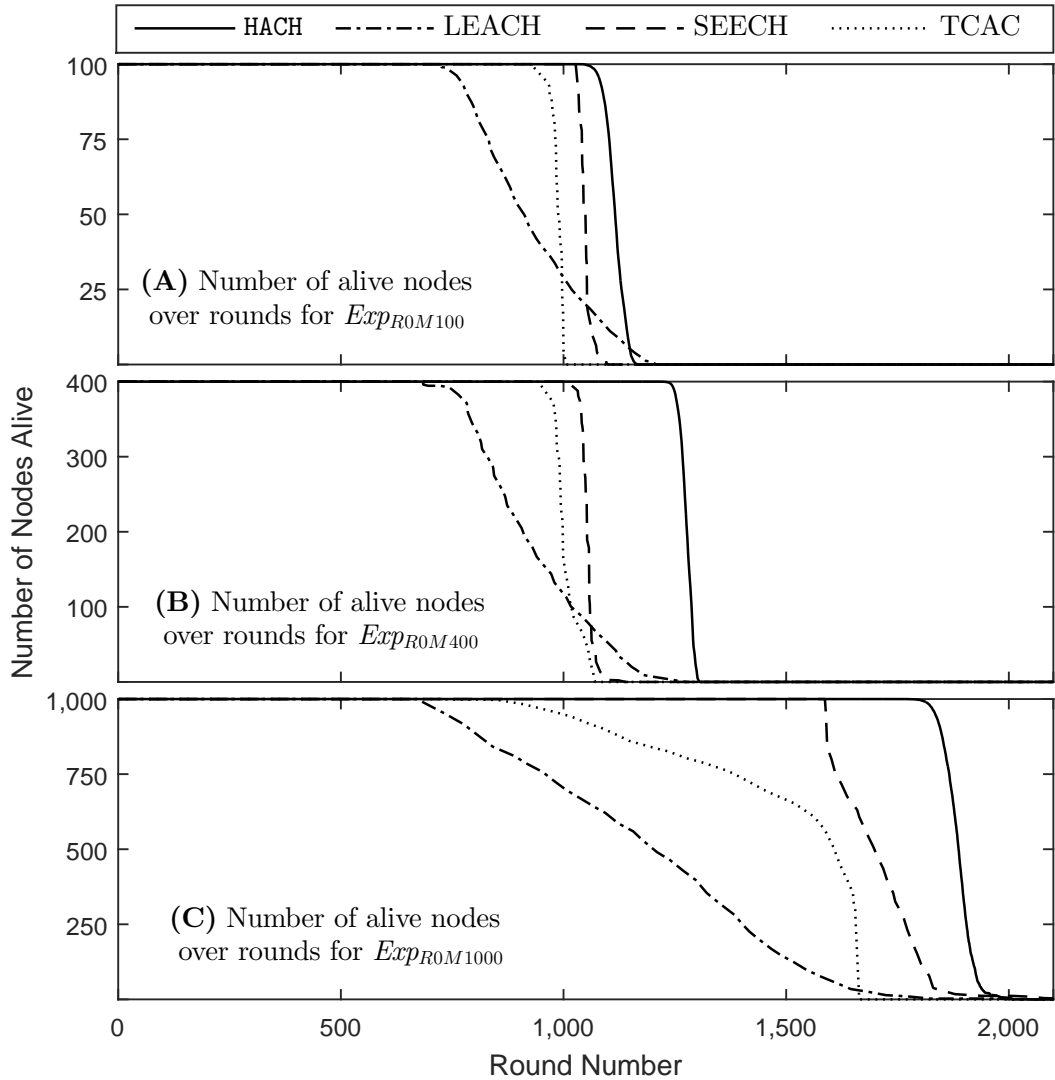


Figure 6.4: Lifetime evaluation of HACH, LEACH, SEECH and TCAC

6. Proposed Global-based Search Algorithm

($Exp_{R0M1000}$) containing 1000 sensors, the proposed algorithm gives extremely high value of 1789 rounds compared with 672, 725 and 1587 round of LEACH, TCAC and SEECH respectively. This shows that as the network size increases, the performance of HACH algorithm continues to improve.

Also, for Experiments Exp_{R0M400} and $Exp_{R0M1000}$ as shown in Figure 6.3, it was deduced that HACH has a very low IPL values for larger network sizes apart from Experiment Exp_{R0M100} which has 30 rounds more than the TCAC protocol. This means that HACH works very well in larger and denser network size. It is also noteworthy that the FND obtained in the proposed HACH protocol for Exp_{R25M0} (See Table 6.5) is 54 rounds more than LEACH protocol (refer to Exp_{R0M100} in Table 6.1); which means that are protocol can still perform with fewer nodes than the LEACH protocol.

6.3.2 Average Energy at First Node Dies (AEFND)

The AEFND is defined as the sum of all current or residual energy values of the sensor nodes divided by the number of nodes at the round when the first node dies. Many nodes begin to die when the first node dies and during the instability periods because of the depleted energy supply. In the HACH protocol, energies of some nodes are balance until the FND time and this is indicated on the graphs of Figure 6.4 by a sharp decline in the number of nodes that are alive for HACH, SEECH and TCAC protocol. One of the performance goals for an energy efficient protocol is to keep the AEFND to a very low value and the proposed HACH protocol kept the AEFND to a very low value of approximately zero for all experiments as shown in Table 6.4 and 6.5. For example, Experiment Exp_{R0M100} has an AEFND of 0.0232J at FND time of 1064 as shown in Figure 6.5.

Table 6.4: AEFND of proposed HACH protocol

	Experiments		
	Exp_{R0M100}	Exp_{R0M400}	$Exp_{R0M1000}$
AEFND	0.0232	0.0164	0.0650

6. Proposed Global-based Search Algorithm

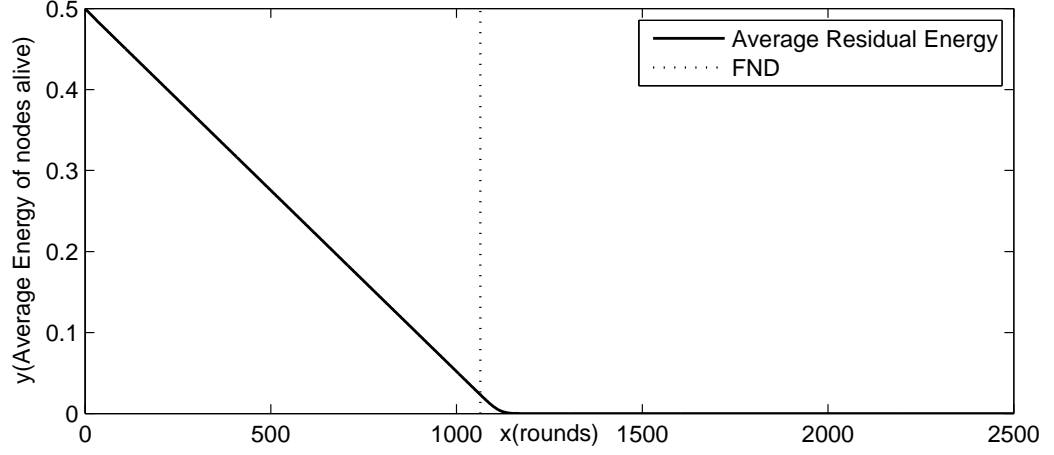


Figure 6.5: Average residual energy of nodes alive versus rounds (refer to Exp_{R0M100})

This proves the fact that sensors were able to manage the energy usage until the FND time. The low AEFND values in Table 6.5 means that the proposed protocol can efficiently manage energy consumption under heterogeneous WSN environments. Therefore, the proposed HACH protocol reduces the energy consumed and enhances energy balance across the nodes in the sensor field thereby extending the network lifespan.

6.3.3 WSNs Heterogeneity

After a certain number of rounds when the sensor networks lifetime has been depleted, new nodes are introduced to re-energise the sensor network. These new nodes are equipped with a higher constant energy value and nodes that are already in use have lower random energy, resulting in energy heterogeneity Kour and Sharma [2010]. As shown in Figure 6.6, the FND value decreases from 1064 for Exp_{R0M100} (refer to Table 6.3) to FND of 780 in Exp_{R25M0} (refer to Table 6.5). Despite the increase in the ratio value of heterogeneous to homogeneous sensors from 25 to 100; which introduces more complexities in terms of energy imbalance, the HACH protocol was still able to balance the energy consumption and maintain a constant FND value.

This phenomenon of starting a network operation with unbalanced energy

6. Proposed Global-based Search Algorithm

Table 6.5: Performance Measures for different heterogeneous WSN Scenarios

Experiment	Performance Measures			
	FND	LND	IPL	AEFND
Exp_{R25M0}	780	937	157	0.040608
Exp_{R25M75}	975	1126	151	0.033479
Exp_{R50M0}	863	1010	147	0.033479
Exp_{R50M50}	976	1061	147	0.030858
Exp_{R75M0}	920	1059	139	0.033468
Exp_{R75M25}	972	1123	151	0.030196
Exp_{R100M0}	971	1110	139	0.033168

distribution in a sensor networks is called WSNs heterogeneity. In this chapter, the experiments that falls under the three level of energy heterogeneity are as follows:

- One-Quarter Level: Experiment Exp_{R25M0} and Exp_{R25M75} .
- Half Level: Experiment Exp_{R50M0} and Exp_{R50M50} .
- Three-Quarter Level: Experiment Exp_{R75M0} and Exp_{R75M25} .

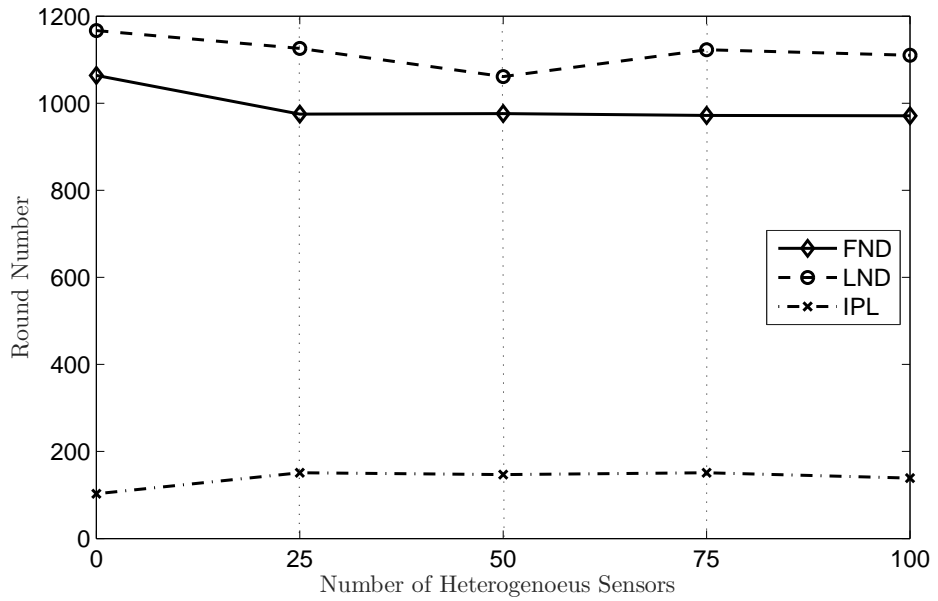


Figure 6.6: Round number versus numbers of heterogeneous sensors

6. Proposed Global-based Search Algorithm

Each level has experiments with Full and Partial heterogeneity. Also, it can be observed in Table 6.5 that adding some energy-homogeneous sensor nodes to a set of energy-heterogeneous or energy depleted sensors extends the lifetime by a considerable amount, for example experiments Exp_{R25M75} , Exp_{R50M50} and Exp_{R75M25} has a FND round of 195, 113 and 52 greater than experiments Exp_{R25M0} , Exp_{R50M0} and Exp_{R75M0} respectively. The performance of each experiment is compared with Exp_{R100M0} , and their percentage value is shown on top of each bar as shown in figure 6.7.

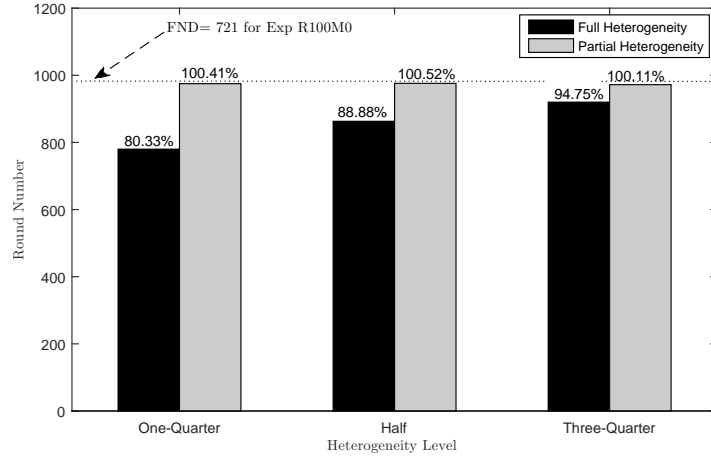
6.3.3.1 Full heterogeneity

Full heterogeneity refers to a scenario whereby all the sensor nodes in a sensing field have random energy values and zero number of constant energy value. Also, this term is used to describe a situation where all sensors in a WSN have different initial energy value. For example in Table 6.2, experiments Exp_{R25M0} , Exp_{R50M0} , Exp_{R75M0} , Exp_{R100M0} are conducted using 25, 50, 75 and 100 number of sensor nodes with random energy values and 0 constant energy values for all the experiments. The bar charts presented in figure 6.7 show that performance improves from one-quarter to the three-quarter full heterogeneity level when compared with Exp_{R100M0} . In figure 6.7a, FND percentages of increasing order of 80.33%, 84.41% and 94.75% were obtained. Also, the LND percentage is in ascending order of 84.41%, 90.99%, 95.41% as shown in figure 6.7b. Additionally the IPL percentage is in decreasing order of 112.95%, 105.76%, 100.0%; meaning the performance increased as the number of heterogeneous nodes increased. Also, in figure 6.7c, Exp_{R50M0} was able to obtain 105.76% which is the same value as the half-level Exp_{R50M50} .

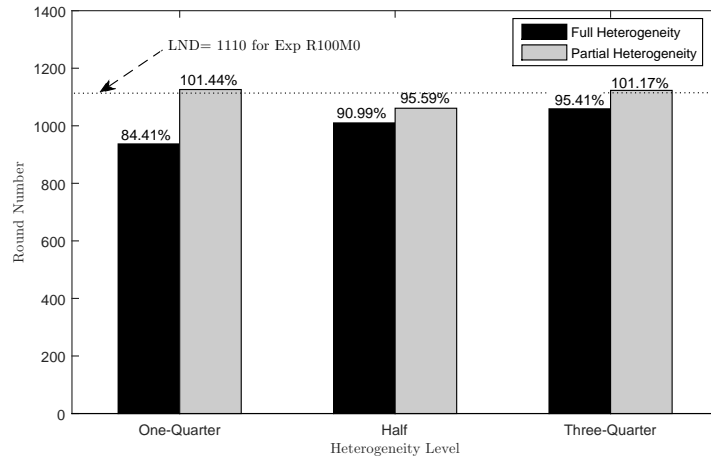
6.3.3.2 Partial heterogeneity

This is the WSN scenario that describes the ratio combination of sensor nodes with random and constant energy values. In Table 6.5, Exp_{R25M75} , Exp_{R50M50} and Exp_{R75M25} use 25, 50, 75 sensor nodes with random energy and 75, 50, 25 sensor nodes with constant energy respectively. In figure 6.7a, the FND time for Exp_{R25M75} , Exp_{R50M50} , and Exp_{R75M25} is 100.41%, 100.52% and 100.11% re-

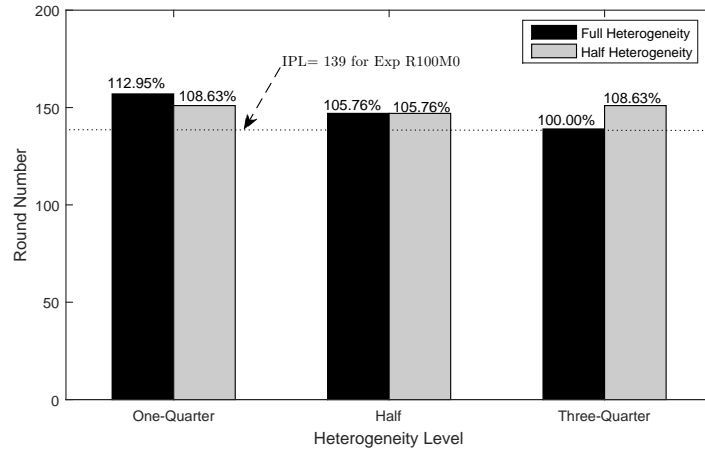
6. Proposed Global-based Search Algorithm



(a) FND



(b) LND



(c) IPL

Figure 6.7: Performance Comparison of different WSNs Heterogeneity Level for (a.) FND, (b.) LND and (c.) IPL measures.

spectively when compared with Exp_{R100M0} ; showing that there is no significant improvement as the ratio of heterogeneous to homogeneous nodes increases. In figure 6.7, Exp_{R50M50} produces the most improved FND of 0.52% more than the Exp_{R100M0} and percentage reduction of LND by 4.41%.

6.4 Conclusion

This chapter proposed a new **HACH** algorithm. The algorithm reduces and balances energy consumption by selecting distributed nodes with high energy as cluster heads to prolong network lifetime. Sequentially, this is achieved by two major operations such as sleep scheduling and cluster head selection operations. The **SSIN** sleep scheduling mechanism inspired by Boltzmann selection process was proposed to decide which nodes to send into sleep mode with negligible effect on the coverage. Subsequently, a genetic algorithm-based technique called the **HEECHS** protocol that would distribute cluster heads evenly within a sensor field to ensure that energy consumption is balanced across the networks is proposed. To guarantee an efficient cluster head selection process, an objective function is designed to evaluate the quality of solutions. Simulation results of the first three experiments shows that the proposed **HACH** algorithm outperforms the **SEECH**, **TCAC** and **LEACH**. This protocol shows a very good performance at large network compared with **DLSACH** protocol. Also, further experiments demonstrated that the proposed protocols can perform even better under different heterogeneity levels of wireless sensor network settings and still maintain acceptable performances.

Chapter 7

Conclusion and Recommended Future Work

In this chapter, the concluding remarks and the recommended future work originating from this research work is presented.

7.1 Thesis Conclusion

WSNs has become an essential component for military and civil applications such as environmental monitoring, target field imaging, weather monitoring, security, battlefield surveillance, event detection etc. Some of the major challenges faced by WSNs are reliability, heterogeneity, scalability and constraint energy supply. The first part of this thesis presented a data aggregation mechanism employed on a multi-dimensional fire data extracted from the sensors in order to remove redundant data. k -means clustering was applied on the aggregated data to form two clusters which represent the two class labels (fire and non-fire). The resulting data outputs are trained by classifiers such as the FFNN, Naive Bayes, and Decision Trees. This hybrid approach of using k -means clustering with classifiers has generated a better fire prediction accuracy against the use of only the classifiers.

Another serious issue with WSNs is energy inefficiency, most especially in large network size. There is need to employ innovative techniques that can eliminate energy inefficiencies which can shorten network lifetime. In the second part of this thesis, the CH selection is seen as an optimisation issue that can be

solved using meta-heuristic and GA approaches. The **DLSACH** and **HACH** protocols were proposed for energy management in wireless sensor networks. An objective function that access the quality of a potential solution was defined for both algorithms. These algorithms perform three sequential operations: sleep scheduling, clustering and energy consumption computation. For the sleep scheduling operation, a **SSIN** mechanism inspired by Boltzmann selection process was proposed to stochastically select some nodes to send into sleep mode without causing an adverse effect on coverage.

The **ILSACHS** mechanism was proposed for clustering operation of the **DLSACH** algorithm. This mechanism employs iterative local search with perturbation operator for cluster head selection. The perturbation operator assists the search process to escape from a local optimum. In addition, it search wider spaces for solution until a quality solution with the better fitness is obtained and no further move is acceptable. After a complete search process, the new solution that guarantees optimally distributed cluster heads is applied to the network. The clustering phase of the **HACH** algorithm employs the proposed **HEECHS** protocol that uses a problem-dependent heuristic crossover to produce a better cluster head configuration that balance energy consumption and enhances well distributed cluster heads. Results shows that the network lifetime of **DLSACH** and **HACH** protocol is more than other protocols such as the **SEECH**, **TCAC** and **LEACH** protocol for 100 and 400 number of sensors. The **DLSACH** protocols shows a poor performance at large network of 1000 sensors with a FND and LND of 183 and 674 rounds respectively. Fortunately, the **HACH** protocol shows tremendous improvement at the same size with FND and LND of 1789 and 2010 rounds respectively.

7.2 Recommended Future Work

The following ideas are recommended to be used for the future:

- Some applications may deploy large number of heterogeneous sensor nodes. Therefore, there is need to test the performance of our approaches under more stringent or complex scenarios such as larger number of sensors with different battery energy value.

-
- A new Memetic-based clustering algorithm that can combine the proposed genetic algorithm with local search operator in order to improve the search strategy towards a better solution that can minimise the energy consumption and extend the lifetime.
 - In addition, a new niching method can be introduced into the new algorithm to minimise the effect of genetic drift caused by selection operator and also prevent the search from being trapped in the local optimum.
 - Different crossover operator types such as uniform, one-point and two-point crossover should be tested in the new algorithm. After several tests, the crossover that produces the best solution should be employed.
 - Tabu Search approach that uses memory and search history can be introduced into the implementation of this new algorithm to avoid double search for a local solution that has been tried and recorded in the tabu lists.

References

- Abbasi, A. A. and Younis, M. [2007], ‘A survey on clustering algorithms for wireless sensor networks’, *Computer communications* **30**(14), 2826–2841. [2](#), [18](#), [69](#)
- Abidha, T. and Mathai, P. P. [2013], ‘Reducing false alarms in vision based fire detection with nb classifier in eadf framework’, *International Journal of Scientific and Research Publications* **3**(8). [50](#), [51](#)
- Aiello, F., Bellifemine, F. L., Fortino, G., Galzarano, S. and Gravina, R. [2011], ‘An agent-based signal processing in-node environment for real-time human activity monitoring based on wireless body sensor networks’, *Eng. Appl. of AI* **24**(7), 1147–1161.
URL: <http://dx.doi.org/10.1016/j.engappai.2011.06.007> [10](#)
- Akkaya, K. and Younis, M. F. [2005], ‘A survey on routing protocols for wireless sensor networks’, *Ad Hoc Networks* **3**(3), 325–349.
URL: <http://dx.doi.org/10.1016/j.adhoc.2003.09.010> [8](#)
- Akthar, F. and Hahne, C. [2012], ‘Rapidminer 5 operator reference’, *Rapid-I GmbH* . [50](#), [65](#)
- Al-Karaki, J. N. and Kamal, A. E. [2004], ‘Routing techniques in wireless sensor networks: a survey’, *IEEE wireless communications* **11**(6), 6–28. [8](#)
- Arboleda, L. M. and Nasser, N. [2006], Comparison of clustering algorithms and protocols for wireless sensor networks, in ‘2006 Canadian Conference on Electrical and Computer Engineering’, IEEE, pp. 1787–1792. [19](#)

REFERENCES

- Bahrepour, M., Meratnia, N. and Havinga, P. J. M. [2009], Use of AI techniques for residential fire detection in wireless sensor networks, *in* ‘Proceedings of the Workshops of the 5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI-2009), Thessaloniki, Greece, April 23-25, 2009’, pp. 311–321.
URL: <http://ceur-ws.org/Vol-475/AIAEP/33-pp-311-321-409.pdf> 60, 65, 66
- Bahrepour, M., Meratnia, N., Poel, M., Taghikhaki, Z. and Havinga, P. J. [2010], Distributed event detection in wireless sensor networks for disaster management, *in* ‘Intelligent Networking and Collaborative Systems (INCOS), 2010 2nd International Conference on’, IEEE, pp. 507–512. 51
- Baskaran, K. [2012], ‘A survey on futuristic health care system: Wbans’, *Procedia Engineering* **30**, 889–896. 10, 11
- Biswas, S. and Morris, R. [2005], ‘Exor: opportunistic multi-hop routing for wireless networks’, *ACM SIGCOMM Computer Communication Review* **35**(4), 133–144. 41
- Borbash, S. A. [2004], Design considerations in wireless sensor networks, PhD thesis, University of Maryland. 44
- Brunda, J., Manjunath, B., Savitha, B. and Ullas, P. [2012], ‘Energy aware threshold based efficient clustering (eatec) for wireless sensor networks’, *Energy* **2**(4). 98
- Bulut, E. and Korpeoglu, I. [2011], ‘Sleep scheduling with expected common coverage in wireless sensor networks’, *Wireless Networks* **17**(1), 19–40. 17
- Cestari, L. A., Worrell, C. and Milke, J. A. [2005], ‘Advanced fire detection algorithms using data from the home smoke detector project’, *Fire Safety Journal* **40**(1), 1–28. 57
- Chachulski, S., Jennings, M., Katti, S. and Katabi, D. [2007], *Trading structure for randomness in wireless opportunistic routing*, Vol. 37, ACM. 41
- Chakraborty, A., Mitra, S. K. and Naskar, M. K. [2011], ‘Energy efficient routing in wireless sensor networks: A genetic approach’, *CoRR abs/1105.2090* . 68

REFERENCES

- Chakraborty, U. K., Das, S. K. and Abbott, T. E. [2012], Energy-efficient routing in hierarchical wireless sensor networks using differential-evolution-based memetic algorithm, *in* ‘2012 IEEE Congress on Evolutionary Computation’, IEEE, pp. 1–8. [41](#)
- Chang, W.-D., Lin, T.-S., Jiang, J.-A., Liu, C.-W., Chen, C.-P., Lai, D.-W., Lu, H.-C., Yen, C.-W. and Yen, P.-L. [2012], ‘An implementation of a wsn-based medical monitoring system: A pilot study of the blood pressure monitoring of hemodialysis patients’, *Engineering in Agriculture, Environment and Food* **5**(3), 83–89. [10](#)
- Charfi, W., Masmoudi, M. and Derbel, F. [2009], A layered model for wireless sensor networks, *in* ‘Systems, Signals and Devices, 2009. SSD’09. 6th International Multi-Conference on’, IEEE, pp. 1–5. [93](#)
- Chen, D. and Varshney, P. K. [2004], Qos support in wireless sensor networks: A survey., *in* ‘International conference on wireless networks’, Vol. 233, pp. 1–7. [44](#)
- Dahnil, D. P., Singh, Y. P. and Ho, C. K. [2012], ‘Topology-controlled adaptive clustering for uniformity and increased lifetime in wireless sensor networks’, *IET Wireless Sensor Systems* **2**(4), 318–327. [20](#)
- Danratchadakorn, C. and Pornavalai, C. [2015], Coverage maximization with sleep scheduling for wireless sensor networks, *in* ‘Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015 12th International Conference on’, IEEE, pp. 1–6. [17](#)
- Deng, J., Han, Y. S., Heinzelman, W. B. and Varshney, P. K. [2005a], ‘Balanced-energy sleep scheduling scheme for high-density cluster-based sensor networks’, *Computer communications* **28**(14), 1631–1642. [16](#), [17](#)
- Deng, J., Han, Y. S., Heinzelman, W. B. and Varshney, P. K. [2005b], ‘Scheduling sleeping nodes in high density cluster-based sensor networks’, *Mobile Networks and Applications* **10**(6), 825–835. [16](#)

REFERENCES

- Dong, Q., Banerjee, S., Adler, M. and Misra, A. [2005], Minimum energy reliable paths using unreliable wireless links, *in* ‘Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing’, ACM, pp. 449–459. [39](#)
- Dounis, A. I. [2010], ‘Artificial intelligence for energy conservation in buildings’, *Advances in Building Energy Research* **4**(1), 267–299. [11](#)
- Dumitrescu, D., Lazzerini, B., Jain, L. and Dumitrescu, A. [2000], *Evolutionary Computation*, International Series on Computational Intelligence, Taylor & Francis. [76](#)
- Eik Loo, C., Yong Ng, M., Leckie, C. and Palaniswami, M. [2006], ‘Intrusion detection for routing attacks in sensor networks’, *International Journal of Distributed Sensor Networks* **2**(4), 313–332. [51](#)
- El Emary, I. M. and Ramakrishnan, S. [2013], *Wireless Sensor Networks: From Theory to Applications*, CRC Press. [26](#)
- Fortino, G., Guerrieri, A., O’Hare, G. M. and Ruzzelli, A. [2012], ‘A flexible building management framework based on wireless sensor and actuator networks’, *Journal of Network and Computer Applications* **35**(6), 1934–1952. [11](#)
- Gen, M., Cheng, R. and Lin, L. [2008], *Network models and optimization: Multiobjective genetic algorithm approach*, Springer Science & Business Media. [27](#)
- Glover, F. [1995], *Tabu search fundamentals and uses*, Graduate School of Business, University of Colorado Boulder. [36](#)
- Glover, F., Kelly, J. P. and Laguna, M. [1995], ‘Genetic algorithms and tabu search: hybrids for optimization’, *Computers & Operations Research* **22**(1), 111–134. [37](#)
- Goldberg, D. E. [1989], ‘Genetic algorithms in search, optimization and machine learning’. [27](#)
- Gorunescu, F. [2011], *Data Mining: Concepts, models and techniques*, Vol. 12, Springer Science & Business Media. [50](#)

REFERENCES

- Gulcharan, N. F. B. I., Daud, H., Nor, N. M., Ibrahim, T. and Shamsudin, M. Z. [2014], ‘Investigation of stability and reliability of the patient’s wireless temperature monitoring device’, *Procedia Computer Science* **40**, 151–159. [10](#)
- Halke, R. and Kulkarni, V. [2012], ‘En-leach routing protocol for wireless sensor network’, *International Journal of Engineering Research and Applications* **2**(4), 2099–2102. [98](#)
- Halkes, G. P., van Dam, T. and Langendoen, K. [2005], ‘Comparing energy-saving mac protocols for wireless sensor networks’, *Mobile Networks and Applications* **10**(5), 783–791. [13](#)
- Hart, W. E., Krasnogor, N. and Smith, J. E. [2005], *Recent advances in memetic algorithms*, Vol. 166, Springer Science & Business Media. [37](#), [38](#), [92](#)
- Hasan, B. S., Khamees, M., Mahmoud, A. S. H. et al. [2007], A heuristic genetic algorithm for the single source shortest path problem, in ‘Computer Systems and Applications, 2007. AICCSA’07. IEEE/ACS International Conference on’, pp. 187–194. [79](#), [97](#)
- Heinzelman, W. B., Chandrakasan, A. P. and Balakrishnan, H. [2002], ‘An application-specific protocol architecture for wireless microsensor networks’, *Wireless Communications, IEEE Transactions on* **1**(4), 660–670. [2](#), [18](#), [25](#), [69](#), [71](#)
- Heinzelman, W. R., Chandrakasan, A. and Balakrishnan, H. [2000], Energy-efficient communication protocol for wireless microsensor networks, in ‘System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on’, IEEE, pp. 10–pp. [18](#)
- Holland, J. H. [1975], *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.*, U Michigan Press. [27](#)
- Hou, J., Fan, X., Wang, W., Jie, J. and Wang, Y. [2010], Clustering strategy of wireless sensor networks based on improved discrete particle swarm optimiza-

REFERENCES

- tion, in ‘2010 Sixth International Conference on Natural Computation’, Vol. 7, IEEE, pp. 3866–3870. [38](#), [41](#)
- Jaafar, K. and Watfa, M. K. [2013], Sensor networks in future smart rotating buildings, in ‘2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)’, IEEE, pp. 962–967. [11](#)
- Jain, A. K. [2010], ‘Data clustering: 50 years beyond k-means’, *Pattern Recognition Letters* **31**(8), 651–666.
URL: <http://dx.doi.org/10.1016/j.patrec.2009.09.011> [53](#)
- James, A. [1999], ‘Using multiple sensors for discriminating fire detection’, *J. FL. USA* pp. 150–164. [57](#)
- Kachitvichyanukul, V. [2012], ‘Comparison of three evolutionary algorithms: Ga, pso, and de’, *Industrial Engineering and Management Systems* **11**(3), 215–223. [viii](#), [28](#), [29](#), [32](#)
- Kasi, M. K., Hinze, A., Legg, C. and Jones, S. [2012], Sepsen: semantic event processing at the sensor nodes for energy efficient wireless sensor networks, in ‘Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems’, ACM, pp. 119–122. [45](#), [46](#)
- Kateretse, C., Lee, G.-W. and Huh, E.-N. [2013], ‘A practical traffic scheduling scheme for differentiated services of healthcare systems on wireless sensor networks’, *Wireless personal communications* **71**(2), 909–927. [10](#)
- Kennedy, J. and Eberhart, R. [1995], Particle swarm optimization, in ‘Neural Networks, 1995. Proceedings., IEEE International Conference on’, Vol. 4, IEEE, pp. 1942–1948. [29](#)
- Kotsiantis, S. B. [2007], ‘Supervised machine learning: A review of classification techniques’, *Informatica (Slovenia)* **31**(3), 249–268.
URL: <http://www.informatica.si/index.php/informatica/article/view/148> [55](#)
- Kour, H. and Sharma, A. K. [2010], ‘Hybrid energy efficient distributed protocol for heterogeneous wireless sensor network’, *International Journal of Computer Applications* **4**(6), 1–5. [104](#)

- Kuila, P. and Jana, P. K. [2014], ‘A novel differential evolution based clustering algorithm for wireless sensor networks’, *Applied soft computing* **25**, 414–425. [42](#)
- Lazarescu, M. T. [2013], ‘Design of a WSN platform for long-term environmental monitoring for iot applications’, *IEEE J. Emerg. Sel. Topics Circuits Syst.* **3**(1), 45–54.
URL: <http://dx.doi.org/10.1109/JETCAS.2013.2243032> [10](#)
- Le, H., Hoang, D. and Poliah, R. [2008], S-web: an efficient and self-organizing wireless sensor network model, *in* ‘International Conference on Network-Based Information Systems’, Springer, pp. 179–188. [17](#)
- Lee, L. C., Nwana, H. S., Ndumu, D. T. and De Wilde, P. [1998], ‘The stability, scalability and performance of multi-agent systems’, *BT Technology Journal* **16**(3), 94–103. [12](#)
- Lewis, F. L. et al. [2004], ‘Wireless sensor networks’, *Smart environments: technologies, protocols, and applications* pp. 11–46. [44](#)
- Liu, J.-L., Ravishankar, C. V. et al. [2011], ‘Leach-ga: Genetic algorithm-based energy-efficient adaptive clustering protocol for wireless sensor networks’, *International Journal of Machine Learning and Computing* **1**(1), 79–85. [71](#)
- Lixin, T. [1999], ‘Improved genetic algorithms for tsp’, *JOURNAL OF NORTH-EASTERN UNIVERSITY (NATURAL SCIENCE)* p. 01. [79](#), [86](#), [96](#), [99](#)
- Mamun, Q. [2012], ‘A qualitative comparison of different logical topologies for wireless sensor networks’, *Sensors* **12**(11), 14887–14913. [95](#)
- Maraiya, K., Kant, K. and Gupta, N. [2011], ‘Efficient cluster head selection scheme for data aggregation in wireless sensor network’, *International Journal of Computer Applications* **23**(9), 10–18. [57](#)
- Mehrotra, K., Mohan, C. K. and Ranka, S. [1997], *Elements of artificial neural networks*, MIT press. [59](#)

REFERENCES

- Memon, I. and Muntean, T. [2012], Cluster-based energy-efficient composite event detection for wireless sensor networks, *in* ‘Sixth International Conference on Sensor Technologies and Applications’, pp. 241–247. [3](#), [55](#)
- Mesin, L., Orione, F. and Pasero, E. G. A. [2011], *Nonlinear Adaptive Filtering to Forecast Air Pollution*, InTech Open Access. [48](#)
- Misra, A. and Banerjee, S. [2002], Mrpc: Maximizing network lifetime for reliable routing in wireless environments, *in* ‘Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE’, Vol. 2, IEEE, pp. 800–806. [40](#)
- Mitchell, T. M. [1999], ‘Machine learning and data mining’, *Communications of the ACM* **42**(11), 30–36. [47](#)
- Mitchell, T. M. [2006], *The discipline of machine learning*, Vol. 9, Carnegie Mellon University, School of Computer Science, Machine Learning Department. [47](#)
- Moscato, P. et al. [1989], ‘On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms’, *Caltech concurrent computation program, C3P Report* **826**, 1989. [37](#)
- Naeimi, S., Ghafghazi, H., Chow, C.-O. and Ishii, H. [2012], ‘A survey on the taxonomy of cluster-based routing protocols for homogeneous wireless sensor networks’, *Sensors* **12**(6), 7350–7409. [1](#), [68](#)
- Nagpal, A., Jatain, A. and Gaur, D. [2013], Review based on data clustering algorithms, *in* ‘Information & Communication Technologies (ICT), 2013 IEEE Conference on’, IEEE, pp. 298–303. [55](#)
- Nie, W., Liu, Y., Li, C. and Xu, J. [2014], ‘A gas monitoring and control system in a coal and gas outburst laboratory’, *Journal of Sensors* **2014**. [10](#)
- Oladimeji, M. O., Turkey, M. and Dudley, S. [2016], A heuristic crossover enhanced evolutionary algorithm for clustering wireless sensor network, *in* ‘Applications of Evolutionary Computation - 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 - April 1, 2016, Proceedings, Part

I', pp. 251–266.

URL: http://dx.doi.org/10.1007/978-3-319-31204-0_17 68, 70

Onwubolu, G. and Davendra, D. [2006], ‘Scheduling flow shops using differential evolution algorithm’, *European Journal of Operational Research* **171**(2), 674–692. 30

Othman, M. F. and Shazali, K. [2012], ‘Wireless sensor network applications: A study in environment monitoring system’, *Procedia Engineering* **41**, 1204–1210. 10

Pal, V., Singh, G. and Yadav, R. [2013], ‘Analyzing the effect of variable round time for clustering approach in wireless sensor networks’, *Lecture Notes on Software Engineering* **1**(1), 31. 96

Park, S., Lee, W. and Cho, D.-h. [2012], Fair clustering for energy efficiency in a cooperative wireless sensor network, *in* ‘Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th’, IEEE, pp. 1–5. 39

Price, K., Storn, R. M. and Lampinen, J. A. [2006], *Differential evolution: a practical approach to global optimization*, Springer Science & Business Media. 33

Puccinelli, D. and Haenggi, M. [2005], ‘Wireless sensor networks: applications and challenges of ubiquitous sensing’, *IEEE Circuits and systems magazine* **5**(3), 19–31. 9

Qing, L., Zhu, Q. and Wang, M. [2006], ‘Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks’, *Computer communications* **29**(12), 2230–2237. 26

Raghunathan, V., Schurgers, C., Park, S. and Srivastava, M. B. [2002], ‘Energy-aware wireless microsensor networks’, *IEEE Signal processing magazine* **19**(2), 40–50. 13

REFERENCES

- Rajagopalan, R. and Varshney, P. K. [2006], ‘Data-aggregation techniques in sensor networks: A survey’, *IEEE Communications Surveys and Tutorials* **8**(1-4), 48–63.
URL: <http://dx.doi.org/10.1109/COMST.2006.283821> 55
- Rajasegarar, S., Leckie, C., Palaniswami, M. and Bezdek, J. C. [2006], Distributed anomaly detection in wireless sensor networks, in ‘2006 10th IEEE Singapore International Conference on Communication Systems’, IEEE, pp. 1–5. 51
- Ramesh, M. V., Kamalanathan, K., Rangan, P. V. et al. [2012], Energy comparison of balanced and progressive sensor networks, in ‘2012 21st Annual Wireless and Optical Communications Conference (WOCC)’, IEEE, pp. 93–98. 16
- Rault, T., Bouabdallah, A. and Challal, Y. [2014], ‘Energy efficiency in wireless sensor networks: A top-down survey’, *Computer Networks* **67**, 104–122.
URL: <http://dx.doi.org/10.1016/j.comnet.2014.03.027> 10
- Saleem, M., Di Caro, G. A. and Farooq, M. [2011], ‘Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions’, *Information Sciences* **181**(20), 4597–4624. 19
- Sangwan, A. and Singh, R. P. [2015], ‘Survey on coverage problems in wireless sensor networks’, *Wireless Personal Communications* **80**(4), 1475–1500. 14
- Shen, C., Srisathapornphat, C. and Jaikaeo, C. [2001], ‘Sensor information networking architecture and applications’, *IEEE Personal Commun.* **8**(4), 52–59.
URL: <http://dx.doi.org/10.1109/98.944004> 55, 79
- Shi, N., Liu, X. and Guan, Y. [2010], Research on k-means clustering algorithm: An improved k-means clustering algorithm, in ‘Third International Symposium on Intelligent Information Technology and Security Informatics, IITSI 2010, Jinggangshan, China, April 2-4, 2010’, pp. 63–67.
URL: <http://dx.doi.org/10.1109/IITSI.2010.74> 53
- Singh, B. and Lobiyal, D. [2013], Energy preserving sleep scheduling for cluster-based wireless sensor networks, in ‘Contemporary Computing (IC3), 2013 Sixth International Conference on’, IEEE. 17

REFERENCES

- Smaragdakis, G., Bestavros, A. and Matta, I. [2004], Sep: A stable election protocol for clustered heterogeneous wireless sensor networks, Technical report, Boston University Computer Science Department. [25](#)
- Souil, M. and Bouabdallah, A. [2011], On qos provisioning in context-aware wireless sensor networks for healthcare, *in* ‘Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on’, IEEE, pp. 1–6. [11](#)
- Sravan, A., Kundu, S. and Pal, A. [2007], Low power sensor node for a wireless sensor network, *in* ‘20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID’07)’, IEEE, pp. 445–450. [13](#)
- Storn, R. and Price, K. [1995], *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces*, Vol. 3, ICSI Berkeley. [30](#)
- Suryadevara, N. K., Mukhopadhyay, S. C., Kelly, S. D. T. and Gill, S. P. S. [2015], ‘Wsn-based smart sensors and actuator for power management in intelligent buildings’, *IEEE/ASME Transactions On Mechatronics* **20**(2), 564–571. [11](#)
- Tan, Z., Liu, Y. and Zhang, Z. [2011], Performance requirements on energy efficiency in wsns, *in* ‘Computer Research and Development (ICCRD), 2011 3rd International Conference on’, Vol. 3, IEEE, pp. 159–162. [33](#)
- Tarhani, M., Kavian, Y. S. and Siavoshi, S. [2014], ‘Seech: Scalable energy efficient clustering hierarchy protocol in wireless sensor networks’, *Sensors Journal, IEEE* **14**(11), 3944–3954. [22](#)
- Tilak, S., Abu-Ghazaleh, N. B. and Heinzelman, W. [2002], ‘A taxonomy of wireless micro-sensor network models’, *ACM SIGMOBILE Mobile Computing and Communications Review* **6**(2), 28–36. [46](#)
- Tubaishat, M. and Madria, S. [2003], ‘Sensor networks: an overview’, *IEEE potentials* **22**(2), 20–23. [12](#)

REFERENCES

- Tyagi, S. and Kumar, N. [2013], ‘A systematic review on clustering and routing techniques based upon leach protocol for wireless sensor networks’, *Journal of Network and Computer Applications* **36**(2), 623–645. [18](#)
- Van Dam, T. and Langendoen, K. [2003], An adaptive energy-efficient mac protocol for wireless sensor networks, in ‘Proceedings of the 1st international conference on Embedded networked sensor systems’, ACM, pp. 171–180. [39](#)
- Vijayvargiya, K. G. and Shrivastava, V. [2012], ‘An amend implementation on leach protocol based on energy hierarchy’, *International Journal of Current Engineering and Technology* **2**(4), 427–431. [71](#)
- Wang, S. and Nie, J. [2010], ‘Energy efficiency optimization of cooperative communication in wireless sensor networks’, *EURASIP Journal on Wireless Communications and Networking* **2010**(1), 1. [37](#)
- Xu, S. and Saadawi, T. [2001], ‘Does the ieee 802.11 mac protocol work well in multihop wireless ad hoc networks?’, *IEEE communications Magazine* **39**(6), 130–137. [39](#)
- Xu, Y., Bien, S., Mori, Y., Heidemann, J. and Estrin, D. [2003], ‘Topology control protocols to conserve energy in wireless ad hoc networks’, *Center for Embedded Network Sensing* . [36](#)
- Xue, C. J. [2010], The road tunnel fire detection of multi-parameters based on bp neural network, in ‘Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on’, Vol. 3, IEEE, pp. 246–249. [48](#)
- Xue, F. and Kumar, P. R. [2004], ‘The number of neighbors needed for connectivity of wireless networks’, *Wireless networks* **10**(2), 169–181. [20](#)
- Younis, O. and Fahmy, S. [2004], ‘Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks’, *Mobile Computing, IEEE Transactions on* **3**(4), 366–379. [25](#)
- Younis, O., Krunz, M. and Ramasubramanian, S. [2006], ‘Node clustering in wireless sensor networks: recent developments and deployment challenges’, *IEEE network* **20**(3), 20–25. [18](#)

REFERENCES

- Yu, L., Wang, N. and Meng, X. [2005], Real-time forest fire detection with wireless sensor networks, in 'Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on'. 48, 56
- Yu, Y., Prasanna, V. K. and Krishnamachari, B. [2006], *Information Processing and Routing in Wireless Sensor Networks*, World Scientific.
URL: <http://www.worldscientific.com/worldscibooks/10.1142/6288> 7, 8
- Zervas, E., Sekkas, O., Hadjiefthymiades, S. and Anagnostopoulos, C. [2007], Fire detection in the urban rural interface through fusion techniques, in 'IEEE 4th International Conference on Mobile Adhoc and Sensor Systems, MASS 2007, 8-11 October 2007, Pisa, Italy', pp. 1–6.
URL: <http://dx.doi.org/10.1109/MOBHOC.2007.4428704> 56
- Zhang, H. [2004], The optimality of naive bayes, in 'Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA', pp. 562–567.
URL: <http://www.aaai.org/Library/FLAIRS/2004/flairs04-097.php> 49
- Zhang, Q. and Sun, J. [2006], Iterated local search with guided mutation, in 'Evolutionary Computation, 2006. CEC 2006. IEEE Congress on', IEEE, pp. 924–929. 78
- Zhiping, L., Huibin, Q. et al. [2006], 'The design of wireless sensor networks for forest fire monitoring system', *School of Electronics and Information, Hangzhou Dianzi University, White Paper* . 56
- Zungeru, A. M., Ang, L. and Seng, K. P. [2012], 'Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison', *J. Network and Computer Applications* **35**(5), 1508–1536.
URL: <http://dx.doi.org/10.1016/j.jnca.2012.03.004> 8, 19

MATLAB Source Codes

A Genetic Algorithm Operators

A.1 Initialization

```
% Function that randomly create an individual with length n argument
.
function X = RandomIndividual(n)% n is the number of sensor node
    X = []; % Create an empty individual set X
    for i=1:1:n % for each sensor node in an individual set X
        if(rand>0.95) % generate a random number and compare with
            0.95
                X(i) = 1; % 1 represent a CH node
        else
            X(i) = 0; % 0 represent a non-CH node
        end
    end
end
end
```

A.2 Tournament Selection

```
% function that select individuals from a population ,pop using a
% tournament size , tn
function [s1, s2] = select_tournament(pop, tn)

    popsize = length(pop); % population size
    s1 = round(rand * (popsize-1))+1; % generate a random point s1
    % within the population size.
    f1 = pop(s1).Fitness; % obtain the fitness value f1 of the
    % individual at the random point s1

    for i= 2:1:tn % loop through the population
        s = round(rand * (popsize-1))+1; % generate a random point s

        while (s1 == s) % If point s1 and s are the same
            s = round(rand * (popsize-1))+1; % generate new random
            point s
        end

        f = pop(s).Fitness; % obtain the fitness value f of the
        % individual at the random point s
        if(f<f1) % If f is less than f1 then,
            f1 = f; % assign fitness value f to f1
            s1 = s; % assign individual s to s1
        end
    end

    s2 = round(rand * (popsize-1))+1; % generate a random point s2
    while (s1 == s2) % If point s1 and s2 are the same
```

```
s2 = round(rand * (popsize-1))+1; % generate new random
    point s2
end
f2 = pop(s2).Fitness; % obtain the fitness value f2 of the
    % individual at the random point s2

for i= 2:1:tn
    s = round(rand * (popsize-1))+1; % generate a new random
        point s
    while (s2 == s || s1 == s) % if s is the same with s2 or s1
        s = round(rand * (popsize-1))+1; % generate another
            point s
    end

    f = pop(s).Fitness; % Obtain fitness value f
    if(f<f2) % if the f is less than f2 then ,
        f2 = f; % assign f to f2 , and
        s2 = s; % assign s to s2
    end
end
end
end
```

A.3 Heuristic Crossover

```
% function that perform heuristic crossover to mate two individuals
% X1 and X2. The third argument n is the number of sensors.
function C = Crossover_Heur(X1, X2, n)
    global SDist2 S2 xm ym; % global variables
```

```
Ch1 = find(X1 == 1); % an array Ch1 that contains CH points of
    X1
Ch2 = find(X2 == 1); % an array Ch2 that contains CH points of
    X2

NewCh = []; % create an empty set NewCh
T = sqrt(xm^2 + ym^2) / (n * 0.05); % compute the threshold
    distance
C = zeros(1,n); % an array C (1 row and n columns) of zero
    elements

AllCh = union(Ch1, Ch2); % union of Ch1 and Ch2

if isempty(AllCh) % if AllCh is empty
    return; %return to previous operation
end

NewCh(1) = AllCh(1); % Assign the first CH point of AllCh to
% the new empty set NewCh
AllCh(1) = []; % Remove the first element in AllCh

while (~isempty(AllCh)) % if AllCh is not empty
    M = SDist2(AllCh(1), NewCh(1)); % Compute the euclidean
        distance
    % between the CH position of AllCh(1) and NewCh(1)
    I = 1;
    for i = 2:length(NewCh)
        if(M > SDist2(AllCh(1), NewCh(i))) % if M is greater
            than
                % euclidean distance between AllCh(1) and NewCh(i)
```

```

        M = SDist2(AllCh(1), NewCh(i)); % Re-compute M
            between
        % AllCh(1) and NewCh(i)
        I = i;
    end
end

if(M < T) % if the value of M is less than threshold
    distance T
    if(S2(NewCh(I)).E < S2(AllCh(1)).E) % if the energy of
        the
        % CH in NewCh is less than energy of CH in AllCh(1)
        NewCh(I) = AllCh(1); % replace NewCh(I) with AllCh
            (1)
    end
else
    NewCh = [NewCh AllCh(1)]; % add AllCh(1) to the elements
        in
        % NewCh and stored in NewCh again
    end

    AllCh(1) = []; % remove the AllCh(1) from the array AllCh
end

C(NewCh) = 1; % insert '1'(CH) into C according on the NewCh
    pointer
end

```


A.4 Mutation

```
% function that mutate the individual X of length n with a
    probability mp
function C = Mutation(X, mp, n)

    C = X; % assign X to C
    if(rand < mp) % if random number is less than mp
        for i=1:1:n
            if(rand<0.005) % if random number is less than 0.005
                C(i) = (C(i)==0); % flip 0 to 1 or 1 to 0
            end
        end
    end

end

end
```

B Proposed SSIN protocol

Coverage

```
% Compute the Coverage using the point covered C, maximum x and y
% field as arguments
function c = Coverage(C, xm, ym)

    s = 0; % initialise s=0
    for x = 1:1:xm
        for y = 1:1:ym
            if(C(x,y) > 0) % point covered C is greater 0
```

```
        s = s + 1; % add point to s variable

    end

end

end

c = s / (xm * ym); % compute the c, given as the total number of
% points covered divided by the grid area

end
```

Send Nodes To Sleep

```
% function that send nodes to sleep mode using argument n only
function SL = SendNodesToSleep(n)

    global Rs C A SumE S1 xm ym; % global variables

    AvgE = SumE / n; % Compute the average energy

    CL = []; % create an empty coverage list CL
    for i = 1:1:n
        if( S1(i).E < AvgE ) % if each sensor's energy is less than
            AvgE
                CL = [CL i]; % add the sensor to coverage list CL
        end
    end

    AA = A(CL); % area covered by the sensor inside the CL
    CC = C; % total point covered by all sensors
    SL = []; % create an empty sleeping list SL

    AccEff = 0; % initialise the accumulated coverage effect to be 0
```

```

MaxPoints = 2 * pi * Rs ^ 2; %circular sensing area covered by
% two sensors

while(~isempty(AA) && AccEff < MaxPoints) % while the AA is not
% empty and AccEff is less than MaxPoints
    [~, I] = min(AA); % find the sensor that covered the
    % smallest area
    % function that update the coverage matrix
    [CC, EE] = UpdateCoverageMatrix(S1(CL(I)), CC, xm, ym);

    % compute probaility P using exponential function of
    % coverage effect ,maximum acceptable coverage effect and
    % accumulated coverage effect
    P = exp( - (EE/MaxPoints) / (1 - (AccEff/MaxPoints))^2 );

    if(rand() < P) % if a random value is less than P
        SL = [SL CL(I)]; % add the sensor in the CL to SL
        AccEff = AccEff + EE; % increment the AccEff by EE
    end

    CL(I) = []; % remove the sensor from the CL
    AA(I) = []; % remove the sensor from the area covered AA
end

end

```

Update Coverage Matrix

```
% function that update the coverage matrix of using sensor in
% coverage list , points covered , xy-maximum xm and ym
function [C1, effect] = UpdateCoverageMatrix( S, C, xm, ym)

    global Rs; % global variable

    x1 = round(S.xd - Rs); % difference between the sensors x
    % coordinate and sensing radius Rs
    if(x1 <= 0) % if x1 is less than or equal to zero
        x1 = 1; % assign 1 to x1
    end

    x2 = round(S.xd + Rs); % add sensors x coordinate plus Rs
    if(x2 > xm) % if x2 is greater than xm
        x2 = xm; % assign xm to x2
    end

    y1 = round(S.yd - Rs); % difference between the sensors y
    % coordinate and sensing radius Rs
    if(y1 <= 0) % if y1 is less than or equal to zero
        y1 = 1; % assign 1 to y1
    end

    y2 = round(S.yd + Rs); % add sensors y coordinate plus Rs
    if(y2 > ym) % if y2 is greater than ym
        y2 = ym; % assign ym to y2
    end

    effect = 0; %initialise effect to be zero
    C1 = C; % assign covered point C to C1
    for x = x1:1:x2
```

```

    for y = y1:1:y2
        % euclidean distance of points covered to sensors
        % is less than or equal to sensing radius Rs
        if( sqrt( (x - S.xd)^2 + (y - S.yd)^2 ) <= Rs) % if
            C1(x,y) = max(0 ,C1(x,y) - 1); % maximum covered
            points
            if(C1(x,y) == 0) % if the covered point is zero
                effect = effect + 1; % increase effect by one
            end
        end
    end
end
end
end
end

```

C Members

```

% function that obtain member sensors from distance SDist ,
% individual X and number of active sensors n arguments
function M= Members(SDist ,X,n)

    if(n==1) % if n is equal to one
        M = 0;
        return; % return the value M=0
    end

    if(n==0) % if n is equal to zero
        M = [];
        return; % return an empty set M
    end % CH→Cluster-head

```

```

CH = find(X == 1); % assign the point at which X=1 to be CH
for i = 1:1:n
    M(i) = 0; % assign zero to point i
    if (X(i) == 0) % if sensor at point i is equal to zero
        m = Inf; I = 0; % m is infinitesimal and I is zero
        for j = 1:1:length(CH) % loop through each CH point
            if (SDist(i,CH(j)) < m) % if euclidean distance
                between
                % sensor point i and CH point j is less than m
                m = SDist(i,CH(j)); % update m
                I = CH(j); % update CH point
            end
        end
        M(i) = I; %assign CH to each sensor
    end
end
end
end

```

D Objective Function

```

% function that computes the fitness values F using argument m and n
function F = ObjectiveFunction(m,n)
    global S2 SumE; % global variable

    if (n==1) % if the number of sensors is equal to one
        F = 1; % assign one to the Fitness F
        return; % exit the function
    end
end

```

```

CHs = find(m == 0); % find the CHs point
L = length(CHs); % length of CH

if(L == 0) % if L is equal to zero
    F = Inf; % F is infinitesimal
    return; % exit the function
end

CHESum = 0; % initialise cluster head sum CHESum
for i = 1:L
    CHESum = CHESum + S2(CHs(i)).E; % Compute CHESum
end

Lower = n * 0.01; % set lower limit to be 1 percent of all
    sensors
Upper = n * 0.03; % set upper limit to be 3 percent of all
    sensors

P = 0; % initialise Penalty risk to be zero
if(L < Lower) % if L is less than lower limit Lower
    P = (Lower - L) / n; % compute P to be (Lower-L)/n
else
    if(L > Upper) % if L is greater than upper limit Upper
        P = (L - Upper) / n; % compute P to be (L-Upper)/n
    end
end

%Compute the Fitness value with weighting factors of 0.7 and 0.3
F = 0.7 * ((SumE-CHESum)/(n-L)) / (CHESum/L) + 0.3 * P;

```

```

    if(isinf(F) || isnan(F)) % if logical array F is equal to one
        F = 1000;
    end
end

```

E Proposed HACH protocol

```

% function that performs heuristic algorithm for clustering hierarchy
function X=HACH(n)
    global popsize STATISTICS r SDist2; % global variables
    STATISTICS.GARounds(r+1) = 0; % obtains statistics for round
    count
    if(n == 0) % if number of sensor is zero
        X=[]; % empty individual
        return; % exit the function
    end

    if(n == 1) % if number of sensor is zero
        X=1; % one individual
        return;
    end

    Counter = 0; % set counter

    S = 0; % initialise S to be zero
    MinF = Inf; % infinitesimal
    MinX = 0; % zero
    MaxF = -Inf; % negative infinitesimal
    MaxX = 0; % zero

```

```

for i=1:1:popsize
    pop(i).Rep=RandomIndividual(n); % generate random individual
    pop(i).M = Members(SDist2, pop(i).Rep, n ); % CH membership
    pop(i).Fitness = ObjectiveFunction(pop(i).M, n); % obtain
    % fitness of individual at point i

    S = S + pop(i).Fitness; % update sum of fitness value S

    if( MinF > pop(i).Fitness) % MinF > individual fitness
        % Assign the individual and its fitness to MinF and MinX
        MinF = pop(i).Fitness;
        MinX = pop(i).Rep;
    end

    if(MaxF < pop(i).Fitness) % MaxF < individual fitness
        % Assign the individual and its fitness to MaxF and MaxX
        MaxF = pop(i).Fitness;
        MaxX = pop(i).Rep;
    end

end

FitAvg1 = S / popsize; %compute the average fitness FitAvg2
FitAvg2 = 0; % initialise FitAvg2 to be zero
FitDiff = abs(FitAvg2 - FitAvg1); % absolute value of difference
% of FitAvg2 and FitAvg1

while (1)
    % initialise MinI, MaxI, MaxX, S to be zero.
    MinI = 0;
    MaxI = 0;

```

```
MaxX = 0;
S = 0;
MaxF = -Inf; % negative infinity
i = 1; % assign one to i
while(i <= popsize)
    % select two points using tournament selection operator
    [p1, p2] = select_tournament(pop,2);
    if (rand < 0.8) %rand <0.80
        % obtain new individual pop2(i) by using heuristic
        % crossover on individual at point p1 and p2
        pop2(i).Rep = Crossover_Heur(pop(p1).Rep, pop(p2).
            Rep, n);
        % Mutate the new individual
        pop2(i).Rep = Mutation(pop2(i).Rep, 0.1, n);
        pop2(i).M = Members(SDist2, pop2(i).Rep, n);
        pop2(i).Fitness = NewFitness(pop2(i).M,n); % Fitness
        % of new individual
        if( MinF > pop2(i).Fitness) % MinF > Fitness
            % assign the individual and its Fitness to MinX and
            % MinF
            MinF = pop2(i).Fitness;
            MinX = pop2(i).Rep;
            MinI = i; % obtain the point i of individual in
            % the population pop2
        end
        if(MaxF < pop2(i).Fitness) % MaxF < Fitness
            % assign the individual and its Fitness to MaxX and
            % MaxF
            MaxF = pop2(i).Fitness;
            MaxX = pop2(i).Rep;
```

```
        MaxI = i;

    end

    S = S + pop2(i).Fitness; % add fitness to S
    i = i + 1; % increase the point i by 1
else % No crossover
    pop2(i) = pop(p1); % assign individual pop(p1) to
        pop2(i)
    pop2(i+1) = pop(p2); % assign individual pop(p2) to
        % next individual to pop2(i)
    % update Sum of fitness
    S = S + pop2(i).Fitness + pop2(i+1).Fitness;
    i = i + 2; % update i

end

end

% Elitism: passing the best (min) individual to the next
% generation Without Selection!!
if (MinI == 0)
    pop2(MaxI).Rep = MinX; % best minimum
    pop2(MaxI).Fitness = MinF; % fitness of the best minimum
end

pop = pop2; % assign pop2 to pop
% update FitAvg2, FitAvg1 and FitDiff
FitAvg2 = FitAvg1;
FitAvg1 = S / popsize;
FitDiff = abs(FitAvg2 - FitAvg1);

Counter = Counter + 1; % increase counter by 1
```

```

    FDiff=MaxF-MinF; % difference between MaxF and MinF

%       if FitDiff < 0.00005 and Counter > 10) or Counter > 100)
    if ((FitDiff < 0.00005 && Counter > 10) || Counter > 100)
        break;
    end
end

STATISTICS.GARounds(r+1) = Counter;
X = MinX; % best individual
end

```

F Proposed DLSACH protocol

```

% function that produces th best individual using the dynamic local
% search algorithm
function X= DLSACH(n)

    global SDist2; % global variable

    X1=RandomIndividual(n); % create a random individual X1
    M = Members(SDist2 , X1, n );
    FitX1= NewFitness(M, n); % Obtain the fitness value FitX1
    counter=0; % initialise counter to be zero
    MaxTrials=100; % Maximum number of trials

    while (counter<=MaxTrials) % counter less than or equal to
        % MaxTrials
    end

```

```
ntimes=round(StepSize(counter,MaxTrails)*n); % Number of
% bits to flip
R=X1; % assign individual X1 to R

for i=1:ntimes
    P=setdiff(round(rand(1)*n),0); % generate random point of
    % bit to flip
    R(P)=(R(P)==0); % flip bit 0 to 1 or 1 to 0
end

Y=Crossover_Heur(X1, R, n); % produce individual Y by using
% heuristic crossover on individual X1 and R
M = Members(SDist2, Y, n );
FitY=NewFitness(M,n); % Compute the fitness FitY

if (FitY<FitX1) % FitY is less than FitX1
    % Assign Y and FitY to X1 and FitX1 respectively
    X1=Y;
    FitX1=FitY;
    counter=0; % reset counter=0
else
    counter=counter + 1; % increase counter by one
end

end

X=X1; % Best individual

end
```

Step Size

```
% function that produces number of bits to flip using counter and
% maximum of trials , MaxTrials arguments
function ntimes = StepSize (counter , MaxTrials)

    range=counter/MaxTrials;

    if (range<=0.5) % <= 1/2 of MaxTrial
        times=0.02; %2 percent
    elseif (range<=0.75) % <= 3/4 of MaxTrial
        times=0.03; %3 percent
    elseif (range<=0.875) % <= 7/8 of MaxTrial
        times=0.04; %4 percent
    else
        times=0.05; %5 percent
    end

    ntimes=times;
end
```

G Energy Consumption

```
% funtion that computes the energy consumption
function EnergyConsumption(m,n)

    global S2 SDist2 BSDist2 k k_CP EDA ; % global variables

    for i = 1:1:n

        % Energy consumed by the member nodes
```

```

    if(m(i) ~= 0)
        E(i) = TXEnergy(k,SDist2(i,m(i))) + 3 * RXEnergy(k_CP) +
            ...
            TXEnergy(k_CP,BSDist2(i));
    else
        M = find(m == i);
        MM = length(M);
        T = 0;
        for j=1:1:MM %TX TDMA to members
            T = T + TXEnergy(k_CP, SDist2(i,M(j)));
        end
        % Energy consumed by the CHs
        E(i) = 2 * RXEnergy(k_CP) + MM *RXEnergy(k) + T + ...
            TXEnergy(k, BSDist2(i)) + TXEnergy(k_CP,BSDist2(i)) +
            ...
            k*EDA*(MM+1);
    end
    S2(i).E = S2(i).E - E(i); % Sensor residual energy
end
end

```

H Main Code

```

global do ETX ERX EDA Efs Emp k k_CP popsize sink S1 S2 Eo...
SDist SDist2 BSDist BSDist2 DistAvg STATISTICS r SumE C Rs A B...
TotalCoverage xm ym; % global variables

% load('Experiment I.mat'); % load files

```

```
% Common Parameters
xm=100; % maximum y value of the sensor field
ym=100; % maximum x value of the sensor field
Rs = 7; % sensing radius
sink.x= 50; % sink x-coordinate
sink.y= 175; % sink y-coordinate
n=75; %The nuber of sensors
popsize=100; % population size
Eo=0.5;
%mu = 0.5; %0.5; % Min Energy for all sensor nodes.
%sigma = 0.05; %0.5; % Max Energy for all sensor nodes.
ETX=50*0.0000000001; % Transmission energy
ERX=50*0.0000000001; % Reception energy
%Transmit Amplifier types
Efs=10*0.00000000000001; % free-space loss
Emp=0.0013*0.00000000000001; % multipath fading
EDA=5*0.0000000001; %Data Aggregation Energy
k = 4000; % Number of packets
k_CP = 50; % Number of control packets
rmax=5000; % Maximum number of rounds

do=sqrt(Efs/Emp); % do computation

% Creation of the Homogeneous Wireless Sensor Network

for i=1:1:n
    S1(i).xd=rand(1,1)*xm; % X coordinates for sensor nodes
    S1(i).yd=rand(1,1)*ym; % Y coordinates for sensor nodes
    S1(i).E=Eo; % assign initial energy to each sensor
```

```

    S1(i).Overlapped = [];
    %initially there are no cluster heads only nodes
    figure (1);
    plot(S1(i).xd,S1(i).yd, 'o'); % plot sensors
    hold on;
end

plot (sink.x,sink.y, '+'); % plot sink

%% Creation of the Full Heterogeneous Wireless Sensor Network
% for i=1:1:n
%     % X and Y coordinates for sensor nodes
%     S1(i).xd=rand(1,1)*xm;
%     S1(i).yd=rand(1,1)*ym;
%     Eo=normrnd(mu,sigma);
%     S1(i).E=Eo; %* (rand + 0.05);
%     S1(i).Overlapped = [];
% end
%
%% Creation of the Partial Heterogeneous Wireless Sensor Network
% for i=1:1:n
%     % X and Y coordinates for sensor nodes
%     S1(i).xd=rand(1,1)*xm;
%     S1(i).yd=rand(1,1)*ym;
%     temp_rnd0=i;
%     Er=Eo;
%     % Randomm Election of Normal Nodes
%     if (temp_rnd0<mdiv*n+1)
%         Er=normrnd(mu,sigma);
%         S1(i).E=Er;

```

```

%         S1(i).Overlapped = [];
%     end
%     % Randomm Election of Advanced Nodes
%     if (temp_rnd0>=mdiv*n+1)
%         S1(i).E=Eo;
%         S1(i).Overlapped = [];
%     end
% end

A = zeros(1,n); % Area covered by the Sensor only A(i)
B = zeros(1,n); % Area within a sensor range covered by others , B(i)

for x = 1:1:xm
    for y = 1:1:ym
        C(x,y) = 0; % Initialise point covered C to be 0
        I = []; % create an empty set I
        for i = 1:1:n
            % if the euclidean distance between point (x,y) and each
            % node coordinate is less than the sensing radius
            if( sqrt( (x -S1(i).xd)^2 + (y - S1(i).yd)^2) <= Rs)
                C(x,y) = C(x,y) + 1; % increase C by 1
                I = [I i]; % add sensor to the array I
            end
        end
        if( ~isempty(I)) % if I is not empty
            if(length(I) == 1) % if I contains one element
                A(I) = A(I) + 1; % area covered by one sensor
                % incremental
            else
                B(I) = B(I) + 1; % area covered by other sensors
            end
        end
    end
end

```

```

                                % within the sensor range incremental
                                end
                                end
                                end
                                end
                                end

% function for computing the Total coverage
TotalCoverage = Coverage(C,xm,ym);

DistSum = 0; % Initialise sum of distances between adjacent
% sensors to be zero
BSDistSum = 0; % Initialise sum of distances between sensors
% and sink
SumE = 0;
for i=1:1:n
    for j=i+1:1:n
        SDist(i,j) = sqrt((S1(i).xd-(S1(j).xd))^2 + ...
            (S1(i).yd-(S1(j).yd))^2); % Compute the euclidean distance
            % between adjacent sensors
        SDist(j,i) = SDist(i,j); % e.g distance between A-B = B-A
        DistSum = DistSum + SDist(i,j); % compute DistSum
    end
    SDist(i,i) = Inf; % infinitesimal distance between sensor and
    % itself
    BSDist(i) = sqrt((S1(i).xd-(sink.x))^2 + ...
        (S1(i).yd-(sink.y))^2); % Compute the euclidean distance between
        % each sensor and the sink
    BSDistSum = BSDistSum + BSDist(i); % compute BSDistSum
    SumE = SumE + S1(i).E; % compute of all the initial energy of
    % all sensors

```

```
end

DistAvg = DistSum / (((n-1)*n)/2); % Compute the average distance
% DistAvg

%%
dead=0; % initialise counter dead sensors
all_dead=0;

allive = n; % number of alive sensors

for r=0:1:rmax

    deadSIndx = []; % creat an empty array , deadSIndx
    SL = SendNodesToSleep(n); % function that decide nodes to sleep
    AL = 1:1:n; % create an active list AL = 1, 2, 3,...,n
    AL(SL) = []; % remove sleeping sensors from the active list AL
    S2 = S1(AL); % update sensors and stored into new variable S2.

    SDist2 = SDist; % assign SDist to SDist2
    % remove euclidean distances of a adjacent sleeping nodes
    SDist2(SL,:) = [];
    SDist2(:,SL) = [];

    BSDist2 = BSDist; % assign BSDist to BSDist2
    BSDist2(SL) = []; % remove euclidean distances of sleeping
    % nodes and sink
    n2 = length(S2); % update the number of active sensors n2

    X = HACH(n2); % function that perform our HACH algorithm
```

```

%      X = DLSACH(n2); % function for DLSACH algorithm
m = Members(SDist2,X,n2);
L = length(find(m == 0)); % Number of CHs
F = ObjectiveFunction(m,n2); % Fitness value of the best
% individual

% Obtain Statistics for the Coverage and Sleeping Nodes
STATISTICS.Coverage(r+1) = UpdateCoverage(TotalCoverage, SL);
STATISTICS.SleepingNodes(r+1)=length(SL);

EnergyConsumption(m,n2); % function that computes the energy
% consumption

S1(AL) = S2; % Update S1

SumE = 0;
for i=1:1:n
    if (S1(i).E<=0) % check if any node is dead
        dead=dead+1; % add dead node to array list dead
        deadSIndx = [deadSIndx i]; % obtain the point of the
            dead node
    else
        SumE = SumE + S1(i).E; % Update the sum of energy SumE
    end
end

% Update the coverage matrix only if sensor is not alive
for i = 1:1:length(deadSIndx)
    [C, aaa] = UpdateCoverageMatrix(S1(deadSIndx(i)),C,xm,ym);
end

```

```
% if the array list deadSIndx is not empty
if(~isempty(deadSIndx))
    TotalCoverage = Coverage(C,xm,ym); % Update the total
    coverage
end

S1(deadSIndx) = []; % Remove dead sensor from the sensor S1

n = length(S1); % obtain new number of sensors n

% Obtain the Statistic of dead sensor DEAD1, alive sensor
% ALLIVE1 and average energy EAvg
STATISTICS.DEAD1(r+1)=dead;
STATISTICS.ALLIVE1(r+1)=allive-dead;
STATISTICS.EAvg(r+1) = SumE / n;

if(isempty(S1)) % if S1 is empty then
    break; % terminate
end

end
```